IMACS '91

13TH WORLD CONGRESS ON COMPUTATION AND APPLIED MATHEMATICS

JULY 22 - 26, 1991
TRINITY COLLEGE DUBLIN IRELAND

PROCEDINGS IN FOUR VOLUMES

VOLUME 2

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited
IMACS '91

Proceedings of the 13th IMACS World Congress on Computation and Applied Mathematics

July 22-26, 1991, Trinity College, Dublin, Ireland

in four volumes

VOLUME 2

Computational Fluid Dynamics and Wave Propagation
Parallel Computing
Concurrent and Supercomputing
Computational Physics/Computational Chemistry and Evolutionary Systems

EDITED BY: R Vichnevetsky
Rutgers University
New Brunswick, USA

J J H Miller
Trinity College
Dublin, Ireland
A UNIFORM NUMERICAL METHOD FOR A CLASS OF QUASILINEAR TURNING POINT PROBLEMS

Relja Vulanović
Institute of Mathematics, University of Novi Sad
21000 Novi Sad, Yugoslavia

Abstract. An $L^1$-stable quasilinear singularly perturbed boundary value problem with a single turning point is solved numerically by a finite-difference scheme on a mesh which is dense near the turning point. The scheme is a special variant of the upwind scheme and it has better properties than the standard Engquist-Osher scheme.

Introduction

We consider the following singularly perturbed boundary value problem:

$$-\varepsilon u''' + \varepsilon b(x, u) u' + c(x, u) = 0, \quad x \in I = [-1, 1],$$

$$u(-1) = U_-, \quad u(1) = U_+,$$

where $\varepsilon$ is a small positive parameter, $U_-$ are given numbers, $b$ and $c$ are sufficiently smooth functions, and

$$c(x, u) = cx_1(x, u) + cx_2(x, u),$$

$$b(x, u) \geq b_0 > 0, \quad x \in I, \quad u \in \mathbb{R}.$$  

Let

$$f(x, u) = \int_0^1 b(x, s) ds, \quad g(x, u) = f_2(x, u) + c(x, u).$$

Then (1) can be written down in the form:

$$-\varepsilon u''' - f(x, u)' + g(x, u) = 0, \quad x \in I.$$  

(5)

Furthermore, we assume:

$$g(x, u) = cx_1(x, u) + (bx(x, u))' \geq g_0 > 0, \quad x \in I, \quad u \in \mathbb{R}.$$  

(6)

Numerical treatment of problems of this type was considered in [2] (the linear case) and [3] (the semilinear case $b = b(x)$). By using the technique from [2], [3], based on inverse monotonicity and (3), (4), (6), we can get that the problem (1), (2) has a unique solution $y$ and that the following estimates hold for $x \in I$:

$$|y(x)| \leq M, \quad |(xy(x))'| \leq M, \quad |(xy(x))''| \leq M[1 + \varepsilon^{-1}u(x)],$$

$$|y''(x)| \leq M|v(x) + \mu + v(x)|, \quad |y'''(x)| \leq M[1 + \varepsilon^{-1}u(x)],$$

where $\mu = \varepsilon^{1/2}$ and $v(x) = \exp(-|x|/\mu)$, and throughout $M$ denotes any positive constant independent of $\varepsilon$. These estimates are needed in the consistency-error analysis.

Numerical Method

We shall use finite-differences on a special discretization mesh, the approach from [2], [3]. The estimates above show that $y$ has an interior layer at $x = 0$, and because of that we shall use a mesh which is dense near that point. The mesh $I^h$ has the points:

$$x_i = \lambda(ih - 1), \quad i = 0(1)n, \quad h = 2/n, \quad n = 2m, \quad m \in \mathbb{N},$$

$$\lambda(t) = \begin{cases} \omega(t) := \mu/1/2 - t, & \text{if } t \in [0, \alpha], \\ \omega'(\alpha)(t - \alpha) + \omega(\alpha), & \text{if } t \in [\alpha, 1], \\ -\lambda(-t), & \text{if } t \in [-1, 0]. \end{cases}$$

Here $(u, w(u))$ denotes the contact point of the tangent line from $(1, 1)$ to $\omega(t)$.

Let $h_i = x_i - x_{i-1}, \ i = 1(1)n$, and let $w^h$ and $z^h$ denote arbitrary mesh functions defined on $I^h \setminus \{-1, 1\}$. We set $w^h = [w_1, w_2, ..., w_{n-1}]^T$ and $w_0 = U_-, \ w_n = U_+$. Furthermore, let

$$D_{-w_i} = (w_i - w_{i-1})/h_i, \quad D_{+w_i} = (w_{i+1} - w_i)/h_{i+1},$$

and let $f_i = f(x_i, w_i), \ g_i = g(x_i, w_i)$. Let us form the discretization of (5), (2):

$$Tw_i := (Tw^h)_i = 0, \quad i = 1(1)n - 1,$$

$$Tw_i = -e[D_{-w_i} - D_{+w_i})/h_i - D_{-f_i} + g_i, \quad i = 1(1)n - 1,$$

$$Tw_i = -e[D_{+w_i} - D_{-w_i})/h_{i+1} - D_{+f_i} + g_i, \quad i = m(1)(n-1) - 1,$$

$$Tw_m = -e[w_{m-1} - 2w_m + w_{m+1})/h_m^2 - [f_{m+1} - f_{m-1}]/2h_m + g_m,$$

(note that $h_m = h_{m+1}$). The following stability inequality holds:

$$\|w^h - z^h\| \leq Mh, \quad \|w_h\| = \sum_{i=1}^m h_i|w_i| + \sum_{i=m+1}^n h_{i+1}|w_i|,$$

and $y_i$ is the unique solution to (7) and $y_h$ is the restriction of $y$ on $I^h \setminus \{-1, 1\}$. Moreover, numerical results show pointwise uniform convergence as well. This is not the case with the Engquist-Osher (EO) scheme [1] in general, see [2]. The EO scheme is uniformly convergent only globally - in the standard discrete $L^1$ norm [1-3]. This is because the EO scheme uses $D_h$ with $(h + h_{i+1})/2$ instead of $h_{i+1}$ and $h_i$. Another advantage of our scheme is that it uses a simpler mesh generating function $\lambda$ than the upwind schemes in [2] and [3].

We illustrate these facts by some numerical results for the problem with $b = 1, \ c = -\pi[x \sin(\pi x) + \varepsilon \pi \cos(\pi x)]$ and $U_- = -2, \ U_+ = 0$, for which the solution is known, see [3]. Let $E$ denote the maximal pointwise error for $\varepsilon = 10^{-12}$. Our scheme gives $E = 0.386, 0.215, 0.115$ for $n = 50, 100, 200$, respectively. Results for other values of $\varepsilon$ are similar due to the special mesh. The EO scheme does not converge: $E = 4.99, 8.131, 8.09$ for the same values of $n$.

References


A NOTE ON A SPLINE COLLOCATION METHOD FOR SINGULARLY PERTURBED PROBLEMS

Katarina Surla
Institute of Mathematics, University of Novi Sad
21000 Novi Sad, Yugoslavia

Zorica Uzelac
Faculty of Technical Sciences, University of Novi Sad
21000 Novi Sad, Yugoslavia

Abstract: The exponential spline collocation methods for singularly perturbed boundary value problems are considered. The convergence between mesh points for different collocation conditions is compared. Numerical results are presented.

Introduction

We consider the following singularly perturbed boundary value problem:

\[ \varepsilon y'' + p(x)y' = f(x), \quad x \in I = [0,1], \]
\[ y(0) = a_0, \quad y(1) = a_1, \]

where \( \varepsilon \) is a small positive parameter, \( a_0 \) and \( a_1 \) are given numbers, \( p \) and \( f \) are sufficiently smooth functions and \( p(x) \geq p > 0 \). By using exponential spline \( e(x) \) from [2], \( e(x) \in C^1(I) \), as a collocation function a family of difference schemes is derived in [4].

The well known Allan-Southwell-Itin and El Mistikawy-Werle (EMW) schemes are members of this family. Some of the properties of the scheme (4),(5),(6), which belongs to the same family, are better than those of EMW scheme (3). In this paper we consider the approximation between the mesh points which correspond to both EMW scheme and to scheme (4),(5),(6). Both splines have first order of uniform convergence on the whole interval, but the numerical results are much better for the scheme (4),(5),(6). This is the consequence of the better accuracy at the nodes. It will also be shown that the collocation spline given in [4] is equal to the one used for the derivation of EMW scheme (1).

Collocation Method

The spline \( e(x) \) has the form (2):

\[ e(x) = c_j(x) = u_j + h m_j t + g_j (c h \mu_j t - 1) / \mu_j + q_j (s h \mu_j t - \mu_j) / \nu_j, \]

where \( t = (x-x_j)/h, \) \( x_j = jh, \) \( h = 1/(n+1) \), \( \mu_j = h \mu_j, \) \( j = 0(1)n, \) \( \nu_j \) are tension parameters. The values \( y_j \) and \( g_j \) are determined from the requirement \( e(x) \in C^1(I) \) from the collocation conditions

\[ e(x) + p^* e'(x) = f^*, \quad x = x_j, x = x_{j+1}, \]

where \( p^* \) and \( f^* \) are constant approximations to \( p(x) \) and \( f(x) \) on the interval \([x_j, x_{j+1}]\) for fixed \( j \), the following family of the difference schemes is derived in [4]:

\[ r^* u_{j-1} + s^* u_j + t^* u_{j+1} = q^* f^* + q^* f^*, \quad j = 1(1)n, \]

where

\[ u_0 = a_0, \quad u_1 = a_1. \]

Numerical results

The \( \varepsilon = 10^{-6} \) is taken from [4]. We denote by \( E_n \) the maximum of \( |y(x_j + h/2) - u_j| + |y(x_j + h/2)| \), \( j = 0(1)n \). The order of convergence (Ord) for two successive values of \( n \) with respective errors \( E_n \) and \( E_{2n} \) is defined in the usual way as in [1]. Tables 1 and 2 present the numerical results obtained by EMW and scheme (4),(5),(6) respectively. The better behaviour for small \( \varepsilon \) of the scheme presented in Table 2 results in hasty decline of Ord.
References


---

![Table 1](table1.png)

![Table 2](table2.png)
A Posteriori Error Bounds for Piecewise Linear Approximate Solutions of Singularly Perturbed Nonlinear Elliptic Problems

Koichi Niijima
Department of Control Engineering and Science
Kyushu Institute of Technology
Iizuka 820, Japan

Abstract: A method for finding a posteriori error bounds for piecewise linear approximate solutions of singularly perturbed nonlinear elliptic problems is proposed. A relation between a line integral on an edge of a triangle and volume integrals in the triangle plays an important role.

1. Introduction
Recently, we developed a method for finding error estimators for piecewise linear approximate solutions of nonlinear elliptic problems (Niijima [1], Niijima [2]). We will apply this method to piecewise linear approximate solutions of singularly perturbed nonlinear elliptic problems. Generally, numerical solutions of such problems do not necessarily have a continuous piecewise linear form. So numerical data obtained are interpolated piecewise linearly such that our method can be applied.

2. Preliminaries
Let $\Omega$ be a bounded polyhedral domain in $\mathbb{R}^2$ with a boundary $\partial \Omega$ and consider the following problem:

$$-\varepsilon \Delta u + f(x, y, u, \nabla u) = 0 \quad \text{in } \Omega, \quad (1)$$
$$u = 0 \quad \text{on } \partial \Omega, \quad (2)$$

where $\varepsilon$ is a sufficiently small positive constant and $\nabla u = (u_x, u_y)$.

A weak form of (1) and (2) is

$$\varepsilon (\nabla u, \nabla v) + (f(x, y, u, \nabla u), v) = 0 \quad (3)$$

for any $v$ belonging to the Hilbert space $H^1_0(\Omega)$, where $(\cdot, \cdot)$ denotes an $L^2(\Omega)$ inner product.

We assume that

II1. (3) has a solution in $H^1_0(\Omega)$,
II2. there exists $\alpha > 0$ such that for $v, w \in H^1_0(\Omega)$,

$$\varepsilon (\nabla (v - w), \nabla (v - w)) + (f(x, y, v, \nabla v) - f(x, y, w, \nabla w), v - w) \geq \varepsilon \|\nabla (v - w)\|^2 + \alpha \|v - w\|^2,$$

where $\|\cdot\|$ indicates an $L^2(\Omega)$ norm.

Consider a triangulation of $\Omega$ and let $F$ be the set of triangles. Let $\tau$ be a triangle in $F$ and let three edges of $\tau$ be $\gamma_1$, $\gamma_2$ and $\gamma_3$. Denote the vertices corresponding to $\gamma_1$, $\gamma_2$ and $\gamma_3$ by $(x_1, y_1)$, $(x_2, y_2)$ and $(x_3, y_3)$, respectively.

We have the following lemma.

Lemma 1. For $g$ belonging to the Hilbert space $H^1(\tau)$, we have

$$(1, g)_{\gamma_3} = \frac{|\gamma_3|}{|J_{\tau}|} (2(1, g)_{\tau} + (x - x_3, g_x)_{\tau} + (y - y_3, g_y)_{\tau}).$$

Here $<\cdot, \cdot>_{\gamma_3}$ and $(\cdot, \cdot)_{\tau}$ indicate inner products on $\gamma_3$ and $\tau$, respectively. Also

$$J_{\tau} = (x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)$$

and $|\gamma_3|$ denotes the length of $\gamma_3$.

This relation is a formula changing a line integral on an edge into the sum of three volume integrals in $\tau$. By this formula, we can rewrite the line integrals appearing in partial integrations of the gradient term by elementwise volume integrals.

3. Main results
Let $E$ be the set of edges not on $\partial \Omega$. Consider
two triangles $\tau_-$ and $\tau_+$ sharing an edge $\gamma$ in $E$, where a normal direction $n$ is outward from $\tau_-$. Let $(x_1, y_1)$, $(x_2, y_2)$ and $(x_-, y_-)$ be the vertices of $\tau_-$, and $(x_1, y_1)$, $(x_2, y_2)$ and $(x_+, y_+)$ the vertices of $\tau_+$. Denote the mesh size by $h$. Let $u^h$ be a continuous piecewise linear function and define a jump in $\partial u^h/\partial n$ across $\gamma$ by

$$\left[\frac{\partial u^h}{\partial n}\right]_{\gamma} = \frac{\partial u^h}{\partial n}|_{r_+} - \frac{\partial u^h}{\partial n}|_{r_-}.$$

For latter convenience, we define $[\partial u^h/\partial n]_{\gamma} = 1$ for edges $\gamma$ on $\partial \Omega$.

Let $\tau$ be a triangle in $F$ and let $\gamma_1$, $\gamma_2$, and $\gamma_3$ be three edges of $\tau$. From now, we use the symbol $u^h$ to denote a piecewise linear interpolate solution of (3). We now define an operator $\Delta^h$ by

$$\Delta^h u^h = \frac{2}{|\gamma_i|} \sum_{i=1}^{3} w_{r,i}|\gamma_i| \left[\frac{\partial u^h}{\partial n}\right]_{\gamma_i}.$$

Here, if $\gamma_i \in E$, then the parameter $w_{r,i}$ has a relation

$$w_{r,i} + w_{r',i'} = 1$$

for the parameter $w_{r',i'}$, where $r'$ is the other triangle sharing $\gamma_i$. If $\gamma_i$ is on $\partial \Omega$, then $w_{r,i}$ is free and $[\partial u^h/\partial n]_{\gamma_i} = 1$.

Using the same symbols as above, we further define a two-dimensional vector $r^h_\tau$ by

$$r^h_\tau = \frac{1}{|\tau|} \sum_{i=1}^{3} w_{r,i}|\gamma_i| \left[\frac{\partial u^h}{\partial n}\right]_{\gamma_i}(x - x_i),$$

$$\sum_{i=1}^{3} w_{r,i}|\gamma_i| \left[\frac{\partial u^h}{\partial n}\right]_{\gamma_i}(y - y_i).$$

We define $\Delta^h$ and $r^h$ by $\Delta^h = (\Delta^h_\tau)_{\tau \in F}$ and $r^h = (r^h_\tau)_{\tau \in F}$, respectively. By $W$, we denote the set of vectors whose components consist of all $w_{r,i}$.

Using Lemma 1, we can prove

Lemma 2. Let $u$ be a solution of (3) and let $u^h$ be a piecewise linear interpolate solution of (3).

We put $e = u - u^h$ and define $L$ by

$$L = (\nabla e, \nabla e) + (f(x, y, u, \nabla u) - f(x, y, u^h, \nabla u^h), e).$$

Then we have

$$L = -(\epsilon \Delta^h u^h + f^h, e) + (e r^h, \nabla e),$$

where we put $f^h = f(x, y, u^h, \nabla u^h)$ for simplicity.

We obtain the following theorem by applying the Schwarz' inequality to the right hand side of $L$ in Lemma 2.

Theorem. We have, for $e = u - u^h$,

$$e \|\nabla e\|^2 + e \| f^h \|^2 \leq \inf_{w} \left\{ \frac{1}{\alpha} - \epsilon \Delta^h u^h + f^h, \alpha \|e\|^2 \right\}. \quad (4)$$

Remark: $\inf_{w} \left\{ \frac{1}{\alpha} - \epsilon \Delta^h u^h + f^h, \alpha \|e\|^2 \right\}$ is a quadratic minimization problem.

4. Numerical results

Example.

$$-\epsilon \Delta u + u^3 + u - g = 0$$

in $\Omega = (0, 1) \times (0, 1)$,

$$u = 0 \text{ on } \partial \Omega,$$

where $\epsilon = 10^{-3}$ and $g$ is determined such that $u = (1 - \exp(-x^2/\epsilon^2))(1 - \exp(-y^2/\epsilon^2))$ satisfies the above equation. It is easily verified that II2 holds as $\alpha = 1$. Divide the interval $(0, 1)$ into $m$-equidistant subintervals and make a triangulation. The numerical solutions $u^h$ were obtained by the Ritz-Galerkin method. A posteriori error bounds were computed following Theorem, and were compared with actual errors.

Using the same symbols as above, we further define a two-dimensional vector $r^h_\tau$ by

$$r^h_\tau = \frac{1}{|\tau|} \sum_{i=1}^{3} w_{r,i}|\gamma_i| \left[\frac{\partial u^h}{\partial n}\right]_{\gamma_i}(x - x_i),$$

$$\sum_{i=1}^{3} w_{r,i}|\gamma_i| \left[\frac{\partial u^h}{\partial n}\right]_{\gamma_i}(y - y_i).$$

We define $\Delta^h$ and $r^h$ by $\Delta^h = (\Delta^h_\tau)_{\tau \in F}$ and $r^h = (r^h_\tau)_{\tau \in F}$, respectively. By $W$, we denote the set of vectors whose components consist of all $w_{r,i}$.

Using Lemma 1, we can prove

Lemma 2. Let $u$ be a solution of (3) and let $u^h$ be a piecewise linear interpolate solution of (3).

We put $e = u - u^h$ and define $L$ by

$$L = (\nabla e, \nabla e) + (f(x, y, u, \nabla u) - f(x, y, u^h, \nabla u^h), e).$$

Then we have

$$L = -(\epsilon \Delta^h u^h + f^h, e) + (e r^h, \nabla e),$$

where we put $f^h = f(x, y, u^h, \nabla u^h)$ for simplicity.
Iterative methods for convection dominated flow*

R. B. Kellogg
University of Maryland
College Park, Md. 20742 USA

Abstract - Some iterative methods are considered for the numerical solution of convection diffusion problems. The first class of iterative methods is Chebyshev accelerated iterations. The issues of parameter selection and convergence rates are considered. Secondly, we consider convection - diffusion type iterations where the iterations are of a Peaceman-Rachford type. Here, a convergence method is established, and a conjecture is given concerning a related problem in-functional analysis.

A. Chebyshev Iterations

We consider Chebyshev accelerated iterations for the numerical solution of discretizations of the convection diffusion equation

$$-\varepsilon A u + p u_{x} + q u_{y} + ru = f \quad \text{in} \, \Omega,$$

and related systems, such as the Oseen system. If a discretized version of (1) is written

$$A u = f,$$

the methods we consider may be written

$$u^{k+1} = a_{k} u^{k} + b_{k} u^{k} + c_{k} u^{k-1},$$

with initial guess \( u^{0} \), where the iteration parameters \( a_{k}, b_{k}, c_{k} \) satisfy

$$b_{k} + c_{k} = 1, \quad c_{k} = 0.$$

From (3) and (4) one finds that the solution \( u \) is preserved under the iteration, and the error \( e^{k} = u - u^{k} \) satisfies

$$e^{k+1} = a_{k} A e^{k} + b_{k} e^{k} + c_{k} e^{k-1},$$

Hence, defining a set of polynomials \( P_{k}(\lambda) \) by

$$P_{0}(\lambda) = 1, \quad P_{k+1}(\lambda) = a_{k} \lambda P_{k}(\lambda) + b_{k} \lambda P_{k}(\lambda) + c_{k} \lambda P_{k-1}(\lambda), \quad k = 0, 1,$$

we find that \( e^{k} = P_{k}(\lambda) e^{0} \). From this formula it is seen that the iteration parameters should be chosen so that the values of \( P_{k} \) are small on the spectrum of \( A \). Manteuffel [1] has shown how to choose the \( P_{k} \) in terms of Chebyshev polynomials so as to optimize the convergence of the iterations. Manteuffel's choice requires a knowledge of an ellipse \( \mathcal{E} \) that contains the spectrum of \( A \) and that does not contain the origin.

In the first part of this talk we show how to obtain ellipses \( \mathcal{E} \) in an explicit manner from a knowledge of the coefficients of the equation (1), the mesh spacing \( h \), and the discretization \( A \). We also give estimates for the asymptotic rate of convergence of the resulting iterative method in terms of the parameters \( \varepsilon \) and \( h \). Finally, we give similar results for a preconditioned version of (1), where we precondition by the self adjoint part of the operator.

Details of this work are contained in [2].

B. Peaceman Rachford type iterations

We consider the convection diffusion equation

$$-\Delta u + p \cdot \nabla u + ru = f, \quad \text{in} \, \Omega,$$

$$u = 0 \text{ on } \Gamma = \partial \Omega.$$

Divide \( \Gamma \) into two subsets, \( \Gamma_{a} = \{(x, y) \in \Gamma : r < 0\} \), and \( \Gamma_{b} = \Gamma \setminus \Gamma_{a} \). Here \( r \) is the outward point unit normal to \( \Gamma \). We consider the CDI method for solving (1): guess \( u^{0} \), and define \( u^{1/2}, u_{+}^{k} \), by

$$-\Delta u^{k+1/2} + p u^{k+1/2} = \rho u^{k} - p \cdot \nabla u^{k} - ru^{k} + f, \quad u^{k+1} = 0 \text{ on } \Gamma_{b},$$

$$p \cdot \nabla u^{k+1} + ru^{k+1} = \rho u^{k+1} - D_{h} u^{k+1} + f, \quad u^{k+1} = 0 \text{ on } \Gamma_{b}.$$

To study the convergence of this method, it is convenient to define the operator \( L_{D} \) by

$$L_{D} u \equiv -\Delta u, \quad u = 0 \text{ on } \Gamma_{b},$$

and the operator \( L_{R} \) by

$$L_{R} u \equiv p \cdot \nabla u + ru, \quad u = 0 \text{ on } \Gamma_{a}.$$ 

With these definitions, the iterations may be written

$$(\rho I + L_{D}) u^{k+1/2} = (\rho I - L_{R}) u^{k} + f,$$

$$(\rho I + L_{R}) u^{k+1} = (\rho I - L_{R}) u^{k+1} + f.$$ 

We regard \( L_{D} \) and \( L_{R} \) as closed, unbounded operators on \( L_{2}(\Omega) \). \( L_{D} \) and \( L_{R} \) are accretive in the sense that for some \( \alpha > 0, \)

$$(L_{D} u, u) > \alpha (u, u),$$

$$(L_{R} u, u) > \alpha (u, u).$$ 

Also, \( L_{D}^{-1} \) and \( L_{R}^{-1} \) are bounded operators on \( L_{2}(\Omega) \), and \( L_{D}^{-1} \) is a compact operator. Finally, any positive \( \rho \) in the resolvent set of \( L_{D} \) and \( L_{R} \). With this understanding we define

$$v^{k+1/2} = (\rho I + L_{D}) u^{k+1/2}, \quad k = 0, 1, \ldots,$$

$$v^{k} = (\rho I + L_{R}) u^{k}, \quad k = 0, 1, \ldots,$$

and we set

$$T_{D} = \quad (\rho I + L_{D}^{-1} (I - L_{D}) = 2\rho (I + L_{D})^{-1} - I,$$

$$T_{R} = \quad (\rho I + L_{R}^{-1} (I - L_{R}) = 2\rho (I + L_{R})^{-1} - I.$$ 

Thus, \( T_{D} \) and \( T_{R} \) are bounded operators, and \( T_{D} \) is a compact perturbation of \( -I \). Also, from the accretiveness of \( L_{D} \) amnd \( L_{D} \) one can... show that

$$||T_{D}|| \leq 1, \quad ||T_{R}|| \leq 1.$$

In terms of these operators, (2) may be written

$$v^{k+1/2} = T_{D} v^{k} + f,$$

$$v^{k+1} = T_{R} v^{k+1/2} + f.$$ 

To establish the convergence of the method, we must show that

$$T_{D} T_{R} v \to 0$$

for any \( v \in L_{2}(\Omega) \). This is easily shown in the finite dimensional case. It becomes an interesting conjecture in the case of the differential operator.

References


---

*Supported in part by the U. S. Army Research Office
EXPLICIT FINITE ELEMENT METHODS FOR
CONVECTION-DIFFUSION PROBLEMS

GERARD R. RICHTER
Department of Computer Science
Rutgers University
New Brunswick, NJ 08903

Abstract. We describe some recent work on explicit finite element methods for convection dominated convection-diffusion problems. We develop the methods for pure hyperbolic equations, and then discuss their extension to problems with diffusion.

1. Introduction. Our purpose is to summarize some recent work on finite element methods for convection-diffusion equations in which convection is the dominant term. One alternative for such problems is the streamline diffusion method [3,4], in which the usual Galerkin's method test functions are augmented by a convective derivative term. There are also "explicit" finite element methods, which permit development of an approximate solution as evolving as a front, in parallel, across layers for a given mesh, and it is potentially advantageous to order the elements to be ordered explicitly with respect to domain. Henceforth we shall deal exclusively with the case of triangular elements.

We need some additional notation to describe these methods. For a generic triangle $T$, let $P_n(T)$ denote the set of polynomials of degree $\leq n$ over $T$, i.e., linear combinations of $x^iy^j, 0 \leq i + j \leq n$. We denote by $S_n^T$ the space of piecewise polynomials over the given triangulation whose restrictions to individual triangles $T$ lie in $P_n(T)$. A function $w_h \in S_n^T$ will in general be discontinuous across triangle edges $T_i$, and for $P \in \Gamma_i$ we define its upstream (−) and downstream (+) limits by $w_h^\pm(P) = \lim_{r \to -a_+} w_h(P \pm a)$. The space $S_n^T \subset S_n^T$ will consist of continuous piecewise polynomials over the same triangulation.

The discontinuous Galerkin method [6,7] produces an approximation $u_h \in S_n^T$ satisfying the conditions

\[
\alpha \cdot \nabla u_h + \beta u_h, v_h \rightleftarrows \int_{\Gamma_{in}(T)} (u_h^+ - u_h^-) u_h \alpha \cdot n \, d\tau = (f, v_h), \quad \forall v_h \in P_n(T).
\]

Here $(\cdot, \cdot)$ is the $L^2(T)$ inner product and $\tau$ denotes arc length along the boundary of $T$. The approximate solution $u_h$ starts off as an interpolant of the given inflow data $g$, and is propagated, triangle by triangle, via the above inner product conditions. The triangles must of course be processed in an explicit order.

To formulate a continuous analog of the discontinuous Galerkin method, we need to distinguish between one-inflow-side (type I) triangles and two-inflow-side (type II) triangles. In developing $u_h \in S_n^T$ in a type I (type II) triangle $T$, note that $u_h$ will have $n+1$ ($2n+1$) fewer degrees of freedom in $T$ because $u_h$, now continuous, is known already on $\Gamma_{in}(T)$. Thus it is natural to define a continuous approximation $u_h \in S_n^T$ via the conditions [7]:

\[
(\alpha \cdot \nabla u_h + \beta u_h, v_h) = (f, v_h), \quad \forall v_h \in V_h,
\]

where $V_h = P_{n-1}(T)$ if $T$ is of type I and $V_h = P_{n-2}(T)$ if $T$ is of type II. This will give equality between the number of equations and unknowns in each triangle. We assume $n \geq 2$ for method (1.3), so that the inner product conditions are nonvacuous for both types of triangles. As before, $u_h$ starts as an interpolant of $g$ on $\Gamma_{in}(\Omega)$.

Both of these finite element methods are generalizations of the most basic first order upwind finite difference scheme.
They share its good stability properties, and may be applied for arbitrarily large $n$. Numerical computations typically show the optimal $O(h^{n+1})$ rate of convergence when the solution $u$ is sufficiently smooth. Theoretical error estimates may be found in [1, 5, 6, 8]. Moreover, the methods yield good results when applied to problems with discontinuous solutions [11]. It can be shown [2] that the influence of a disturbance propagating along a characteristic is confined to a band of width approximately $O(h)$ about the characteristic. Thus as $h \to 0$, the methods methods exhibit the right limiting domain of dependence behavior.

2. Convection-diffusion equations. We now consider a convection-diffusion equation

$$(2.1) \quad -\epsilon \Delta u + \alpha \cdot \nabla u + \beta u = f, \quad x \in \Omega, \
$$

$$u = g, \quad x \in \Gamma,$$

where $\alpha$, $\beta$ before, has unit length, and $\epsilon$ is a positive constant. If $\epsilon$ is large in comparison to the mesh size $h$, then diffusion will be the dominant transport term, and the standard Galerkin method performs well. However, if $\epsilon$ is small, convection will be dominant, and Galerkin’s method is no longer the finite element method of choice. Solution features that are not resolved generate oscillations which tend to propagate throughout the domain. The methods (1.2) and (1.3) can be extended to convection dominated problems of the form (2.1), thus providing an alternative.

The discontinuous Galerkin method can be extended to (2.1) by treating the diffusion term in essentially the same way as the convection term:

$$(2.2) \quad (-\epsilon \Delta u_h + \alpha \cdot \nabla u_h + \beta u_h, v_h) + \int_{\Gamma_{1h}(T)} \left( \frac{\partial u_h}{\partial n} - \frac{\partial u_{h,k}}{\partial n} \right) v_h \, dr$$

$$- \int_{\Gamma_{2h}(T)} (u_h^+ - u_h^-) v_h \alpha \cdot n \, dr = (f, v_h), \quad \text{all } v_h \in P_n(T).$$

This scheme uses upstream values of both $u_h$ and its normal derivative. (A minor modification needs to be made if $\Gamma_{1h}(T) \cap \Gamma_{2h}(\Omega) \neq 0$ because $\frac{\partial u}{\partial n}$ is available only in the interior of $\Omega$.) The approximation $u_h$ starts as an interpolant of $g$ on $\Gamma_{1h}(\Omega)$, and is developed in the same explicit manner as in the pure hyperbolic case. It is shown in [12] that this scheme is stable for sufficiently small $\frac{\epsilon}{\alpha}$, assuming the triangle sides (with the possible exception of those on $\Gamma$) are uniformly bounded away from the characteristic direction. Near optimal error estimates are also derived there. Similar extensions of the continuous method (1.3) are developed in [9, 10].

For a convection-diffusion problem in which convection is dominant only over part of $\Omega$, these hyperbolic-based finite element methods can be applied locally, in conjunction with the standard Galerkin method. For example, the solution to (2.1) typically has an outflow boundary layer of thickness $O(\epsilon)$ [13]. One could apply (2.2), say, up to this point and then use Galerkin’s method over a finer mesh to resolve the outflow boundary layer.

**References**


ON THE DETERMINATION OF THE ORDER OF UNIFORM CONVERGENCE

Paul A. Farrell
Department of Mathematics & Computer Science
Kent State University
Kent, OH 44242, U.S.A.

Alan Hegarty
Department of Mathematics
University of Limerick
Limerick, Ireland.

Abstract: We shall discuss a number of methods used in the literature to calculate rates of uniform convergence. We mention some anomalies concerning interpretation of the resulting tables and rates, which lead to the determination of experimental rates of uniform convergence lower than the correct rates.

Introduction

The determination of the order of uniform convergence is not always a straightforward task. A number of approaches exist in the literature, the two major variants being that appearing in [6] and [1] and that in [2].

The former approach involves solving numerically a singularly perturbed differential equation on \([z_L, z_R]\) for which the analytic solution is known. The difference equation is solved for decreasing values of \(h\) and the rate of convergence calculated from

\[
p_{\text{cr}}(\epsilon) = \frac{\ln e_{\text{cr}}^h - \ln e_{\text{cr}}^h}{\ln 2}
\]

where

\[
e_{\text{cr}}^h = \max_{0 \leq \epsilon \leq \epsilon^*} |u_\epsilon^h - u_{\epsilon}(x)|, \quad h = [z_L - z_R]/N.
\]

The equation solved is chosen so that the solution and its derivatives exhibit exactly the analytic behaviour hypothesized in the proof of the error estimates. In practice this is achieved by choosing a solution and then determining a differential equation of the correct form which this satisfies.

The uniform rate is determined by inspecting a table of values of \(p_{\text{cr}}\) for varying \(h\) and \(\epsilon\), constructed by setting \(\epsilon = \epsilon^*\) for various values of \(\epsilon\). Results of this form are given in [6] for a non-turning point problem and in [1] for a turning-point problem of the type considered in [3,4]. In the case of the turning-point problem the choice of \(\epsilon = \epsilon^*\) functions of \(h\). A disadvantage of this method is the requirement of prior knowledge of the solution since this limits the ease with which it can be applied to other problems. An alternative is, of course, to use an accurate approximation on a fine mesh, if this is determinable.

The Double Mesh Method

The method proposed in [2] is based on a consequence of the General Convergence Principle which states

Theorem ([2, Theorem I.5.1])

Let \(u_\epsilon\) be the solution of a differential equation and \(u_\epsilon^h\) a difference approximation. Let \(p > 0\) and \(C_1, C_2\) be positive constants independent of \(h\). Then, for all \(i \geq 0, 0 < h < h_0, \) and all \(\epsilon > 0\)

\[
|u_i(x) - u_\epsilon^h| \leq C_1 h^p
\]

if

(i) \(\lim_{h \to 0} |u_i(x) - u_\epsilon^h| = 0\)

(ii) \(|u_i^h - u_\epsilon^h| \leq C_2 e^{-\alpha h}\).

Furthermore \(C_2\) is independent of \(\epsilon\) iff \(C_2\) is.

Essentially the method involves calculating the quantity given in (ii) which we shall call the double mesh error \(e_{\text{cr}}^h\), and determining a rate of convergence

\[
p_{\text{cr}}(\epsilon) = \frac{\ln e_{\text{cr}}^h - \ln e_{\text{cr}}^h}{\ln 2}
\]

where

\[
e_{\text{cr}}^h = \max_{0 \leq \epsilon \leq \epsilon^*} |u_\epsilon^h - u_{\epsilon}(x)|, \quad h = [z_L - z_R]/N.
\]

The experimental uniform rate of convergence is then determined as

\[
p = \min_{p_{\text{cr}}} \text{ where } p_{\text{cr}} = \text{average}_{p_{\text{cr}}}
\]

Doolan, Miller and Schilders remark that the choice of the range of \(h\) values permissible is limited, since if \(h\) mesh is too coarse the solution of the difference scheme is not sufficiently representative of the solution of the differential equation to permit meaningful discussion of convergence, that is \(h\) is not sufficiently small, whereas if it is too fine then rounding error predominates. The method has the advantage that it requires no a priori knowledge of the nature of the solution of the equation and may be easily programmed to determine an experimental rate of convergence for a wide variety of problems.

Anomalies

In either of these methods however, great care must be taken in interpreting the value of tables of \(p_{\text{cr}}\) or \(p_{\text{cr}}\) for the reasons which we will outline below. To simplify the arguments we shall consider only the rate of convergence for a non-turning point problem, as considered in [6], that is:

\[
w(x) + a(x)u(x) - b(x)u = f(x), \quad 0 < x < 1
\]

and

\[
0 \leq \epsilon < 1
\]

where \(a(x) \geq \delta > 0\). The determination of the rate of convergence depends on the assumption that

\[
e_{\text{cr}}^h < C h^p
\]

where \(C\) is independent of \(h\) and \(\epsilon\). This is not necessarily the strongest bound available and in fact the following one, (cf. [6]), is a more accurate estimate:

\[
e_{\text{cr}}^h < C(h^2 + \delta) e^{-\alpha h},
\]

where \(\rho = h/\epsilon\) and \(\alpha > 0\). So more accurately

\[
e_{\text{cr}}^h < C(\rho, \alpha, h).
\]

which is not inconsistent with (6) since by considering the limits as \(\rho \to 0\) and \(\rho \to \infty, \) for a fixed \(h,\) we can see that \(C(\rho, \alpha, h)\) is bounded. Let us assume that (6) holds in \(\delta,\) and determine the rate of convergence \(p_{\text{cr}}\). Thus

\[
p_{\text{cr}} = p_{\text{cr}}(\alpha, h) = \ln \left(\frac{4h_1 \epsilon^2 e^{-\alpha h}}{(1 + \rho)^{1 - \epsilon} + e^{-\alpha h}}\right) - \ln 2
\]

and considering this for \(\epsilon = \alpha = 0\) and \(\epsilon = \alpha = \rho \to \infty\) we get

\[
\lim_{\epsilon \to 0} p_{\text{cr}} = 2, \quad \lim_{\epsilon \to \infty} p_{\text{cr}} = 1.
\]

Thus, if we determine the rate of convergence \(p_{\text{cr}}\) for fixed \(h,\) and \(\epsilon\) varying from 0 to 0, it will vary from 2 to 1 as expected. The assumption we implicitly make in evaluating the experimental orders of convergence is that \(p_{\text{cr}}\) is independent and hence that the minimum value is 1. This is not in fact the case as the function \(p_{\text{cr}}\) given by (9) may attain a minimum less than 1. This may be seen in Fig. 1, which is a graph of \(p_{\text{cr}}\) for \(\epsilon = 0.05\). The lack of monotonocity of \(p\) is, in fact, most apparent when \(\epsilon\) is small. Similar results are also given in Fig. 1 for \(p_{\text{cr}}\).
the rate of convergence might be expected from comparison of the two curves in Fig. 1. For example, the minima are calculated using the double mesh method that are less prone to this effect, although these will again report lower than actual rates of uniform convergence. We remark however that it does not give any additional information about the variations in behaviour of the scheme as \( h \to 0 \) or \( \epsilon \to 0 \). Therefore it is also useful to include tables of \( p_{\rho} \) to provide more precise details of this behaviour. This is particularly so for problems having more complicated boundary or interior layers.

We should remark at this point that there remain certain problems. In particular, the restriction that \( h \) was sufficiently large is crucial. If \( h \) becomes small, the most prevalent effect is for rounding errors to corrupt the results. However, if the calculations were done "exactly", so that rounding error were absent or negligible, then a more serious problem would arise. If we produce tables for arbitrarily small \( h \), but only for finitely small \( \epsilon \), then most of the rates in the table will be for \( h \ll \epsilon \). In this case, we are in the region where classical convergence theory applies and thus the rates will be greater than or equal to 1, for most schemes. These rates will dominate the table, and, if we use \( p_{\rho} \) or \( p_{\rho} \) as the calculated rate, cause even non-uniform schemes to be reported as uniformly convergent. This may be viewed as a consequence of the form of the tables, where, in this case in particular, the rate of convergence is a function of \( \rho = h/\epsilon \). Thus the rates along the diagonals are equal. To get an accurate reflection of the rate of uniform convergence it is therefore necessary to extend the table at least as far in \( \epsilon \) as in \( h \). In the one-dimensional cases, which we have tested, (cf. [3]), we extend the table until the errors, for given \( h \), stabilize, which occurs when one is solving, up to rounding error, the reduced equation. The finest mesh used in the calculations is \( h = 1/4096 \). In practice, using either double or fine mesh methods this has given acceptably accurate rates. In all cases the rate calculated using the fine mesh method proved higher. A more cautious approach might be to use \( p_{\rho} \) or \( p_{\rho} \), which are less prone to this effect, although these will again report lower than actual rates of uniform convergence.

We remark that there are other circumstances in which these methods will report positive uniform convergence rates where, using the normal definition, the scheme would not be considered uniformly convergent. This is particularly true of the centered difference approximation to a self-adjoint problem and of two or higher dimensional problems exhibiting certain phenomena. These issues are considered further in [5].

It is clear that \( e^h_{\rho} \) is a function of \( \epsilon \) alone and thus we may expect \( p_{\rho} \) to be approximately a constant independent of \( h \). This will lead to a better estimate for the uniform rate of convergence. In the case \( \alpha = 0.25 \), this is \( p_{\rho}^1 = p_{\rho}^2 = p_{\rho}^3 = 1.00 \). The \( p_{\rho} \)-method has been used to determine the rate of convergence in [3, 4] and many later-papers. We remark however that it does not give any additional information about the variations in behaviour of the scheme as \( h \to 0 \) or \( \epsilon \to 0 \). Therefore it is also useful to include tables of \( p_{\rho} \) to provide more precise details of this behaviour. This is particularly so for problems having more complicated boundary or interior layers.

We should remark at this point that there remain certain problems. In particular, the restriction that \( h \) was sufficiently large is crucial. If \( h \) becomes small, the most prevalent effect is for rounding errors to corrupt the results. However, if the calculations were done "exactly", so that rounding error were absent or negligible, then a more serious problem would arise. If we produce tables for arbitrarily small \( h \), but only for finitely small \( \epsilon \), then most of the rates in the table will be for \( h \ll \epsilon \). In this case, we are in the region where classical convergence theory applies and thus the rates will be greater than or equal to 1, for most schemes. These rates will dominate the table, and, if we use \( p_{\rho} \) or \( p_{\rho} \) as the calculated rate, cause even non-uniform schemes to be reported as uniformly convergent. This may be viewed as a consequence of the form of the tables, where, in this case in particular, the rate of convergence is a function of \( \rho = h/\epsilon \). Thus the rates along the diagonals are equal. To get an accurate reflection of the rate of uniform convergence it is therefore necessary to extend the table at least as far in \( \epsilon \) as in \( h \). In the one-dimensional cases, which we have tested, (cf. [3]), we extend the table until the errors, for given \( h \), stabilize, which occurs when one is solving, up to rounding error, the reduced equation. The finest mesh used in the calculations is \( h = 1/4096 \). In practice, using either double or fine mesh methods this has given acceptably accurate rates. In all cases the rate calculated using the fine mesh method proved higher. A more cautious approach might be to use \( p_{\rho} \) or \( p_{\rho} \), which are less prone to this effect, although these will again report lower than actual rates of uniform convergence.

We remark that there are other circumstances in which these methods will report positive uniform convergence rates where, using the normal definition, the scheme would not be considered uniformly convergent. This is particularly true of the centered difference approximation to a self-adjoint problem and of two or higher dimensional problems exhibiting certain phenomena. These issues are considered further in [5].

References


Abstract. In this paper, numerical examples are presented, which indicate that upwinded finite difference schemes on special meshes are numerically e-uniformly convergent for the numerical solution of elliptic singular perturbation problems; for these examples it is also shown that upwinded finite difference schemes on uniform meshes behave unsatisfactorily. The numerical results given here validate the theoretical results obtained by the last author in [3, 4].

1. INTRODUCTION

In this paper, the following linear singular perturbation problem

\begin{equation}
Lu \equiv \varepsilon \Delta u + \alpha \nabla u + a_0 u = f \quad \text{on} \quad \Omega = (0,1) \times (0,1) \quad (1.1a)
\end{equation}

\begin{equation}
u = g \quad \text{on} \quad \partial \Omega \quad (1.1b)
\end{equation}

where \( \alpha = (\alpha_1, \alpha_2) \) and \( 0 < \varepsilon \leq 1 \) is examined. Two cases are considered:

\begin{align}
\alpha_1 & \geq \alpha_1, \alpha_2 \geq \alpha_2 \quad \text{and} \quad \alpha_1 + \alpha_2 > 0 \quad (1.1c) \\
\alpha_1 & \geq \alpha_1, > 0, \quad \alpha_2 \equiv 0 \quad (1.1d)
\end{align}

where \( \alpha_1, \alpha_2 \) are constants and \( \alpha_1, \alpha_2, a_0, f \) and \( g \) are smooth enough to ensure no interior or corner layers.

In each case, for small values of \( \varepsilon \), boundary layers appear near some of the sides of \( \Omega \). In the case (1.1a,b,c) regular layers appear near \( x = 0 \) and \( y = 0 \) and in the case (1.1a,b,d) a regular layer appears near \( x = 0 \) and parabolic layers near the sides \( y = 0 \) and \( y = 1 \) which lie along characteristics of the reduced differential equation \( \varepsilon = 0 \). For small values of \( \varepsilon \), it is well known that classical numerical methods for (1.1) may produce spurious oscillations throughout the whole domain.

Various stable upwind methods have been proposed to eliminate these oscillations. In §2, numerical examples demonstrate that the nodal errors for an upwinded method of this type (on a uniform mesh) depend not only on the number of mesh elements used but also on the value of \( \varepsilon \). Numerical results for this section suggest that the error is essentially uniform for small enough \( \varepsilon \) and that the number of layers increases with \( \varepsilon \). More precisely, a numerical method for solving (1.1) is \( \varepsilon \)-uniformly convergent of order \( p \) on the mesh \( \omega_\varepsilon = \{(x_i, y_j): i, j = 0, 1, \ldots, N\} \) if

\begin{equation}
\max_{\omega_\varepsilon} || u - z || \leq C N^{\varepsilon - p}; \quad (1.2)
\end{equation}

where \( u \) is the solution of (1.1a,b,c) or (1.1a,b,d), \( z \) is the numerical approximation to \( u \), \( C \) and \( p > 0 \) are independent of \( \varepsilon \) and \( N \) is the number of mesh elements used.

The following upwinded finite difference operators will always be used, to obtain a numerical approximation \( z \) to \( u \):

\begin{align}
\varepsilon (\delta_{x+i,1} + \delta_{x+1,1}) z + \alpha_1 (x_i, y_j) \delta_{x} z + a_2 (x_i, y_j) \delta_{y} z \\
+ a_0 (x_i, y_j) z = f(x_i, y_j) \quad (1.3)
\end{align}

where

\begin{align}
\delta_{x+i,1} z & \equiv \frac{z(x_{i+1}, y_j) - z(x_i, y_j)}{h_i+1} \\
\delta_{x+1,1} z & \equiv \frac{z(x_i, y_{j+1}) - z(x_i, y_j)}{h_j} \\
\delta_{x} z & \equiv \frac{z(x_{i-1}, y_j) - z(x_i, y_j)}{h_i} \quad \text{and} \quad h_i \equiv (h_{i+1} + h_i)/2,
\end{align}

on various meshes \( \omega_\varepsilon \equiv \{(x_i, y_j)\} \).

Shishkin [4] has proved that on a uniform mesh no \( \varepsilon \)-uniform finite difference scheme exists for a problem with a parabolic layer, such as (1.1a,b,d). In §3 a method which is \( \varepsilon \)-uniform for all types of layers is described. This method was first introduced by Shishkin in [3] and uses classical upwinded difference operators on a special piecewise-uniform mesh. Numerical results are presented for specific examples of (1.1a,b,c) and (1.1a,b,d).

2. Classical upwinding on a uniform mesh

In this section, the numerical performance of the upwinded difference scheme (1.3) on the uniform mesh

\[ \omega_E \equiv \{(x_i, y_j): i, j = 0, 1, \ldots, N\}, \]

where \( h = 1/N \) and \( N \) is the number of mesh elements used in both directions, is examined when applied to examples of (1.1a,b,c) and (1.1a,b,d). The problems are solved on a sequence of meshes, with \( N = 8, 16, 32, 64, 128 \). The errors \( |z(x_i, y_j) - u(x_i, y_j)| \) are approximated for successive values of \( \varepsilon \) on the four coarsest meshes by

\[ E_{\varepsilon} = \max_{i,j} |z(x_i, y_j) - z^{(24)}(x_i, y_j)|, \]

where the superscript indicates the number of mesh elements used. For each \( \varepsilon \) the maximum nodal error is approximated by

\[ E_{\varepsilon,N} = \max_{i,j} |z^{(2)}(x_i, y_j)|. \]

Convergence rates are calculated for each \( \varepsilon \) and for \( N = 8, 16, 32 \), are estimated by

\[ p_{\varepsilon,N} = \log_2 \left( \frac{E_{\varepsilon,N}}{E_{\varepsilon,2N}} \right). \]

These estimated rates are given in tables 2.1 and 2.2 for the following problems:

Problem 2.1: \( \varepsilon \Delta u + (2 + x^2 y)u_x + (1 + xy)u_y = 0 \) on \((0,1) \times (0,1)\), with boundary conditions:

\begin{align}
u(x, 0) & = 0; \quad u(x, 1) = \begin{cases} 4x(1 - x), & x > 1/2, \\ 1, & x \leq 1/2 \end{cases} \\
u(0, y) & = 0; \quad u(1, y) = \begin{cases} 8(y - 2y^2), & y > 1/4, \\ 1, & y \leq 1/4 \end{cases}
\end{align}

Problem 2.2: \( \varepsilon \Delta u + (1 + x^2 + y^2)^2u_x = 0 \) on \((0,1) \times (0,1)\), with boundary conditions

\begin{align}u(x, 0) & = 0; \quad u(x, 1) = x^2; \quad u(0, y) = 0; \quad u(1, y) = 1. \end{align}

Problem 2.1 has regular layers near the sides \( x = 0 \) and \( y = 0 \); the estimated convergence rates are:
Problem 2.2 has a regular layer near the side \( z = 0 \) and parabolic layers near \( y = 0 \) and \( y = 1 \); the estimated convergence rates are:

<table>
<thead>
<tr>
<th>( \varepsilon )</th>
<th>( N=8 )</th>
<th>( N=16 )</th>
<th>( N=32 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.000000000000</td>
<td>0.82</td>
<td>1.10</td>
<td>1.54</td>
</tr>
<tr>
<td>0.500000000000</td>
<td>0.92</td>
<td>1.13</td>
<td>1.54</td>
</tr>
<tr>
<td>0.250000000000</td>
<td>0.86</td>
<td>1.07</td>
<td>1.52</td>
</tr>
<tr>
<td>0.125000000000</td>
<td>0.84</td>
<td>0.97</td>
<td>1.46</td>
</tr>
<tr>
<td>0.062500000000</td>
<td>0.46</td>
<td>0.98</td>
<td>1.37</td>
</tr>
<tr>
<td>0.031250000000</td>
<td>-0.07</td>
<td>0.66</td>
<td>1.35</td>
</tr>
<tr>
<td>0.015625000000</td>
<td>-0.59</td>
<td>0.17</td>
<td>1.11</td>
</tr>
<tr>
<td>0.007812500000</td>
<td>-0.20</td>
<td>-0.40</td>
<td>0.69</td>
</tr>
<tr>
<td>0.003906250000</td>
<td>0.18</td>
<td>-0.24</td>
<td>0.08</td>
</tr>
<tr>
<td>0.001953125000</td>
<td>0.30</td>
<td>0.21</td>
<td>-0.13</td>
</tr>
<tr>
<td>0.000976562500</td>
<td>0.35</td>
<td>0.42</td>
<td>0.25</td>
</tr>
<tr>
<td>0.0004882813</td>
<td>0.38</td>
<td>0.55</td>
<td>0.66</td>
</tr>
<tr>
<td>0.00002441406</td>
<td>0.40</td>
<td>0.62</td>
<td>0.89</td>
</tr>
<tr>
<td>0.00001220703</td>
<td>0.40</td>
<td>0.67</td>
<td>1.06</td>
</tr>
<tr>
<td>0.00000610352</td>
<td>0.41</td>
<td>0.69</td>
<td>1.17</td>
</tr>
</tbody>
</table>

In fact, it is well known: that upwinding on a uniform mesh is not uniformly in \( \varepsilon \) convergent; however, in neither case is an estimate of the uniform convergence rate calculated because the maximum value of \( N \) is not large enough to make such estimates meaningful.

3. Numerical results on special meshes.

For the problem (1.1a,b,c), define the special mesh \( \omega_{1,2}^* \) by

\[
\omega_{1,2}^* = \{(x_i, y_j) : 0 \leq i, j \leq N \}
\]

where

\[
x_i = h_1, \quad \text{for } 0 \leq i \leq N/2,
\]

\[
x_i = \sigma_2 + (i - (N/2))h_2, \quad \text{for } N/2 \leq i \leq N,
\]

\[
h_1 = 2\sigma_2/N, \quad \text{and } \quad h_2 = 2(1 - \sigma_2)/N.
\]

The points \( \{y_j^*\} \) are defined analogously. The transition point \( \sigma_2 \) is chosen to depend on both the layer width and the number of mesh elements used.

\[
\sigma_2 \equiv \min\left(1/2, C_1 \varepsilon \ln N\right), \quad C_1 > \alpha_1
\]

The transition point \( \sigma_2 \) is defined analogously. Using the upwinded difference operator (1.3) on this special mesh, we obtain the following estimates of convergence rates for problem 2.1:

For the problem (1.1a,b,d), define the special mesh \( \omega_{2,3}^* \) by

\[
\omega_{2,3}^* = \{(x_i, y_j^*) : 0 \leq i, j \leq N \}
\]

where

\[
x_i = 2\alpha_2/N, \quad i = 0, 1, \ldots, N/2,
\]

\[
x_i = \sigma_2 + 2(i - N/2)(1 - \sigma_2)/N, \quad i = N/2, \ldots, N,
\]

\[
y_i = 4\varepsilon \sigma_2/N, \quad i = 0, 1, \ldots, N/4,
\]

\[
y_i = \sigma_2 + 2(i - N/4)(1 - \sigma_2)/N, \quad i = N/4, \ldots, 3N/4,
\]

\[
y_i = (1 - \sigma_2) + 4(i - 3N/4)(\sigma_2)/N, \quad i = 3N/4, \ldots, N,
\]

where \( \sigma_2 \equiv \min\left(1/4, \varepsilon \ln N\right) \), and \( \sigma_2 \equiv \min\left(1/4, \sqrt{\varepsilon \ln N}\right) \).

Using (1.3) on this mesh, we obtain the following convergence rate estimates for problem 2.2:

<table>
<thead>
<tr>
<th>( \varepsilon )</th>
<th>( N=8 )</th>
<th>( N=16 )</th>
<th>( N=32 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.000000000000</td>
<td>1.71</td>
<td>1.53</td>
<td>1.38</td>
</tr>
<tr>
<td>0.250000000000</td>
<td>1.02</td>
<td>1.19</td>
<td>1.56</td>
</tr>
<tr>
<td>0.062500000000</td>
<td>0.92</td>
<td>1.05</td>
<td>1.40</td>
</tr>
<tr>
<td>0.015625000000</td>
<td>0.91</td>
<td>1.07</td>
<td>1.43</td>
</tr>
<tr>
<td>0.003906250000</td>
<td>0.77</td>
<td>1.01</td>
<td>1.47</td>
</tr>
<tr>
<td>0.000976562500</td>
<td>0.78</td>
<td>1.01</td>
<td>1.48</td>
</tr>
<tr>
<td>0.0004882813</td>
<td>0.81</td>
<td>0.98</td>
<td>1.46</td>
</tr>
<tr>
<td>0.0002441406</td>
<td>0.82</td>
<td>0.97</td>
<td>1.44</td>
</tr>
<tr>
<td>0.0001220703</td>
<td>0.83</td>
<td>0.97</td>
<td>1.43</td>
</tr>
<tr>
<td>0.0000610352</td>
<td>0.83</td>
<td>0.97</td>
<td>1.43</td>
</tr>
</tbody>
</table>

While, again, no estimate of the uniform convergence rate is given for either problem, the rates for each \( \varepsilon \) and \( h \) are evidently far superior to those obtained using a uniform mesh. A fuller discussion of the numerical experiments outlined above can be found in [1],[2].

References

CONSTRUCTION AND ANALYSIS OF PETROV-GALERKIN APPROXIMATIONS FOR CONVECTION-DOMINATED FLOWS

B.W. SCOTNEY
Department of Mathematics, University of Ulster
Cromore Road, Coleraine, BT52 1SA.

Abstract - Error estimates are shown for Galerkin and Petrov-Galerkin approximations to a singularly perturbed problem in one dimension. An optimal Petrov-Galerkin formulation is presented for variable coefficient problems, and it is used to analyse the common practice of approximating variable coefficient problems by the use of locally constant methods.

1. INTRODUCTION
We consider the model problem

\[-au''(x) + b(x)u'(x) = f(x), \quad x \in (0,1)\]

\[u(0) = g_L, \quad u(1) = g_R \tag{1.1}\]

where \(a > 0\), \(b(x) \in H^1\) with \(b(x) > 0\) and \(f(x) \in L_2\).

For the case when \(b\) is a positive constant, Hemker (1977) developed a Petrov-Galerkin formulation with a piecewise linear trial space which generates a nodally exact solution. Here we develop an optimal Petrov-Galerkin approximation with a piecewise linear trial space for the general variable velocity problem (1.1).

2. WEAK FORMULATION AND ERROR ESTIMATES
The weak formulation of problem (1.1) is to find \(u \in H^1_{0,0}\) such that

\[B(u,v) = \langle f,v \rangle \quad \forall v \in H^1_{0,0} \tag{2.1}\]

where \(B(\cdot,\cdot)\) is the bilinear form

\[B(w_1,w_2) = \langle w_1',w_2' + bw_1 \rangle \]

For a piecewise linear trial space \(S^h \subset H^1\) and \(S^h_{0,0} = S^h \cap H^1_{0,0}\), the Galerkin approximation is to find \(U \in S^h_{0,0}\) such that

\[B(U,V) = \langle f,V \rangle \quad \forall V \in S^h_{0,0} \tag{2.2}\]

If we denote by \(\|w\|_{T_1}^h\) the norm on \(H^1_{0,0}\) defined by

\[\|w\|_{T_1}^2 = \langle w',w' \rangle \tag{2.3}\]

the Lax-Milgram Theorem (see, for example, Ciarlet (1978)), together with a Friedrichs-Poincaré inequality yields the following error estimate for the Galerkin approximation \(U: \)

\[\|u-u\|_{T_1}^h \leq (1 + b/\text{sa}) \inf_{v \in S^h_{0,0}} \|u-v\|_{T_1} \tag{2.4}\]

where \(b = \max_{x \in (0,1)} |b(x)|\).

If an Aubin-Hitsche duality argument (see Aubin (1972)) is used, an improved estimate can be achieved of the form

\[\|u-u\|_{T_1}^h \leq (1 + bh/\text{sa}) \inf_{v \in S^h_{0,0}} \|u-v\|_{T_1} \tag{2.5}\]

In Scotney (1985) the optimal estimate

\[\|u-u\|_{T_1} \leq (1 + (bh/\text{sa})^{1/2}) \inf_{v \in S^h_{0,0}} \|u-v\|_{T_1} \tag{2.6}\]

is obtained and shown to be the sharpest attainable bound.

The estimates (2.5) and (2.6) exhibit the importance of the mesh Péclet number \(bh/\text{sa}\) in the degree to which optimality of the Galerkin approximation is lost as the differential operator loses self-adjointness. Petrov-Galerkin methods may be formulated for singular perturbation problems with a view to recapturing the optimal approximation properties enjoyed by the Galerkin approximation for self-adjoint problems. A test space \(Th \in C^1\) other than \(S^h\) is employed. Setting \(Th = Th \cap H^1_{0,0}\), the Galerkin system (2.2) is replaced by the problem of finding \(U \in S^h_{0,0}\)

such that \(B(U,V) = \langle f,V \rangle \quad \forall V \in Th_{0,0}\) \tag{2.7}\)

Norton (1982) shows how to achieve bounds sharper than those obtained directly from the Lax-Milgram Theorem, such as (2.4). He identifies the crucial requirements to establish optimal error estimates. By the kiesz Representative Theorem (see, for example, Adams (1975)), there exists a map

\[R: H^1_{0,0} \rightarrow Th_{0,0} \text{ with } \inf_{j \in \mathbb{N}} \|U-R\|_{T_1} \leq \inf_{V \in Th_{0,0}} \|U-V\|_{T_1} \tag{2.8}\]

If the constant \(\Delta\) is defined by

\[\Delta = \sup_{v \in Th} \inf_{V \in Th_{0,0}} \|v-V\|_{T_1} \tag{2.9}\]

and \(U\) is the Petrov-Galerkin solution to problem (2.7) the following estimate holds:

\[\|u-u\|_{T_1} \leq (1 - \Delta^2)^{-1/2} \inf_{v \in Th_{0,0}} \|u-v\|_{T_1} \tag{2.10}\]

This is the sharpest possible estimate since there will exist a function \(f \in L_2\) for which (2.10) is an equality. (2.8), (2.9) and (2.10) provide the basis for Section 4.

J. PETROV-GALERKIN METHODS FOR CONSTANT COEFFICIENT PROBLEMS
We consider a uniform discretisation with nodes \(x_j = jh, \quad j = 0,\ldots,N, \quad \text{and } Th = h = 1\).

For the problem (1.1) when \(b > 0\) is a constant the test space \(th_{0,0}\) proposed by Hemker (1977) has a basis \(\{\varphi_j^h, j = 1,\ldots,N-1\}\) with \(\varphi_j^h(x) = e^{-\beta(x-x_j-1)/h} - 1/(e^{-\beta} - 1), \quad \varphi_j = \chi_j \chi_{j-1} \chi_{j+1}
\[e^{-\beta(x-x_j)/h}/(e^{-\beta} - 1), \quad \varphi_j = \chi_j \chi_{j+1} \chi_{j+2} \in \mathbb{N}\]
\[0, \quad \text{otherwise}\]
where $\beta = bh/a$. The solution of (2.7) thus generated is nodally exact. It is straightforward to show that the best fit $U^* \in S_E^h$ to

$$u \in H^1_E$$

in the norm $\| \cdot \|_{T_1}$ is also nodally exact, and hence that $U = U^*$.

Numerous Petrov-Galerkin methods have been proposed for problem (1.1) which generate the same finite difference operator as the test space defined by (3.1), namely that of Allen & Southwell (1955) - see, for example, Heinrich & Zienkiewicz (1979). Such methods are not optimal since they differ in their treatment of the source term $f$. However Scotney (1985) has used the estimate (2.10) of Morton (1982) to show that many are near-optimal. Griffiths & Lorenz (1978) have analysed these methods by direct use of the Lax-Milgram Theorem.

4. RIESTZ REPRESENTATION

In the variable coefficient problem (1.1), from the defining relation (2.8) we may deduce that

$$(R^w)(x) = w(x) + \int_0^X (b(t)/a)w(t)\,dt - x\alpha(w)$$

$$v w \in H^1_E$$

(4.1)

where $\alpha(w) = \int_0^1 (b(t)/a)w(t)\,dt$ (4.2)

From (2.9) and (2.10) it is clear that if

$$R^w_E = S_E^h$$

then the Petrov-Galerkin solution is optimal. We therefore aim to construct a basis of $\{\psi_j : j = 1, \ldots, N-1\}$ such that $R\psi_j \subset S^h$ with the additional property that the support of $\psi_j$ is restricted to the support of the piecewise linear trial space basis function $\phi_j$, namely $(x_j, x_{j+1})$. The key to the localisation is to set $R\psi_j$ to be of the same form as

$$R\psi_j = 0$$

In particular let

$$(R\psi_j)(x) = \psi_j(x) + \int_0^X (b(t)/a)\psi_j(t)\,dt - x\alpha_j$$

$$v \psi \in H^1_E$$

(4.3)

where $\alpha_j = \int_0^1 (b(t)/a)\psi_j(t)\,dt$ (4.4)

Writing $g(x) = (R\psi_j)(x) + \alpha_j x$

$$g(x) = \psi_j(x) + \int_0^X (b(t)/a)\psi_j(t)\,dt - x\alpha_j$$

(4.5)

we obtain

$$\psi_j(x) = e^{-X(b(t)/a)\,dt} \int_0^X g'(s)e^{X(b(t)/a)\,dt}ds$$

(4.6)

From Scotney (1985) we take the required form for $g(x)$ as below:

$$0$$

(4.7)

$$T_j/h(x - x_j)$$

for $x_j < x \leq x_{j+1}$

The explicit form of $\psi_j(x)$ may be obtained by substituting (4.7) into (4.6), satisfying the requirement for local support and normalisation. Restricting the support of $\psi_j$ to that of $\phi_j$ requires us to set $\psi_j(x) = 0$ for $x_{j+1} < x$,

yielding $T_j = \frac{\alpha_j}{\phi_j} n_j$ (4.8)

where $n_j = \int_0^{X_k} e^{b(t)/a}dt, k = 0, \ldots, N$ (4.9)

and $B(x) = \int_0^x (b(t)/a)\,dt$ (4.10)

Normalising by setting $\psi_j(x_j) = 1$ gives

$$w(x) = \frac{h(y_j - y_{j-1})e^{b(x_j)}}{\gamma_j}$$

(4.11)

and hence we obtain $\psi_j(x) = e^{X(b(t)/a)\,dt} \int_0^X g'(s)e^{X(b(t)/a)\,dt}ds, x_j < x \leq x_{j+1}$

(4.12)

It is straightforward to check that if $b(x) = b > 0$, a constant, (4.12) reduces to the test function $\psi_j$ of Hemker (1977) as in (3.1).

If the test space generated by (4.12) is used in the Petrov-Galerkin formulation (2.7) it can be shown that the system of linear equations generated is of the form

$$a_j \psi_j = h(y_j - y_{j-1})e^{b(x_j)}$$

$$\psi_j(x_j) = 1$$

(4.13)

The significance of (4.13) for the variable coefficient problem (1.1) is analogous to that of the Allen & Southwell (1955) difference operator for the constant coefficient problem.

5. LOCALLY CONSTANT METHODS FOR VARIABLE COEFFICIENT PROBLEMS

It has been commonplace for many authors to generalise the use of Hemker's test space (or its near-optimal counterparts) to variable coefficient problems by selecting the value of $b$ in $\phi_j$ to be determined locally at each node or in each element (e.g. $b = b(x_j)$ for $x \in (x_{j-1}, x_{j+1})$, or $b = (b(x_{j-1}) + b(x_j))/2$ for $x \in (x_{j-1}, x_j)$). Only with the availability of $\psi_j(x)$ as in (4.12) is it possible to properly analyse this practice.

Consider a modified form of problem (2.1) in which the velocity field $b(x)$ is replaced by a piecewise constant field $B(x)$ defined locally on each element: find $u \in H^1_E$ such that
\[ \mathbf{B}(u,v) = \langle f,v \rangle \quad \forall v \in H^1_{E_0} \quad (5.1) \]

where \( \mathbf{B}(x) = E_j = \frac{b(x_j) + b(x_{j+1})}{2} \)

for \( x \in (x_j, x_{j+1}) \),

\( j = 0, \ldots, N-1 \), and \( \mathbf{B}(\cdot, \cdot) \) is \( B(\cdot, \cdot) \) with \( b(x) \) replaced by \( \mathbf{B}(x) \).

If we denote by \( \mathbf{V}_j \) the test function \( \mathbf{V}_j^H \) of Hemker with \( b(x) \) replaced by \( \mathbf{B}(x) \), we may consider the modified Petrov-Galerkin formulation: find \( \mathbf{U} \in \mathcal{S}_{E} \) such that

\[ \mathbf{B}(\mathbf{U}, \mathbf{V}_j) = \langle f, \mathbf{V}_j \rangle \quad j = 1, \ldots, N-1 \quad (5.2) \]

The formulation (5.2) is an obvious generalisation to the variable coefficient problem (1.1) of the use of Hemker's constant coefficient test space described in (3.1).

By substituting \( \mathbf{B}(x) \) for \( b(x) \) in (4.12), we can show that the optimal test space for problem (5.1) is precisely the one given by \( \mathbf{V}_j \) in (5.2). That is, the obvious generalisation of Hemker's formulation generates the optimal approximation to the solution of the modified problem (5.1).

Moreover, \( \mathbf{V}_j(x) \) may be written in the form

\[ \mathbf{V}_j(x) = \begin{cases} \phi_j(x) + \xi_{j-1} \sigma_{j-1}(x) & x_j < x < x_{j-1} \\ \phi_j(x) - \xi_j \sigma_j(x) & x_j < x < x_{j+1} \end{cases} \quad (5.3) \]

where

\[ \xi_j \sigma_j(x) = 0 \]

and

\[ \int_{x_j}^{x_{j+1}} \sigma_j(x) \, dx = \frac{h}{2} \quad (5.4) \]

Then the system of linear equations generated by (5.2) may be written as

\[ \begin{align*}
-a/h - (1 + \xi_{j-1}) \mathbf{E}_{j-1}/2 & \ U_{j-1} \\
+ (2a/h + ((1 + \xi_{j-1}) \mathbf{E}_{j-1} - (1 - \xi_j) \mathbf{E}_j)/2) & \ U_j \\
+ (-a/h + (1 - \xi_j) \mathbf{E}_j/2) & \ U_{j+1} = \langle f, \mathbf{V}_j \rangle \\
\end{align*} \quad (5.5) \]

Since (5.4) implies no knowledge of the functional form of \( \sigma_j(x) \), any of the conforming Petrov-Galerkin test spaces used for the constant coefficient problem can be modified to fit (5.3) and hence used to generate the left hand side of (5.5).
The system of differential equations describing the behaviour of semiconductor devices is singularly perturbed, and therefore requires special discretisation techniques. To this end, an exponentially fitted box method has been developed, which is known as the Scharfetter-Gummel method. In this paper, we will discuss this discretisation technique, which is applicable to n-dimensional problems. Application of this technique is not restricted to semiconductor device problems, it is suitable for any singularly perturbed problem in divergence form. It is conjectured that this method yields uniform error estimates for arbitrary polygonal meshes. It is also shown that the 1-d scheme is essentially the same as Iln's scheme, thus paving the way for uniform error estimates on the solution of the semiconductor problem. On the other hand, the way the scheme has been constructed is different from the construction of Iln's scheme, and therefore this may be of interest to singular perturbationists. Finally, extensions of the scheme to a special case are discussed, where exponentially fitted methods for small systems of equations can be used. Again it is suspected that uniform error estimates may be obtained.

\[ J_p = -q\mu_n n_{\text{int}} e^{(\phi_s - \phi) / \mu_r} \nabla \phi_p \]  
\[ J_n = q\mu_n n_{\text{int}} e^{(\phi_s - \phi) / \mu_r} \nabla \phi_n \]

A third choice of variables can be motivated by remarking that the system of equations can be put into self-adjoint form, which is convenient in designing suitable discretisation schemes. This can be achieved by applying a Liouville transformation. This result is the formulation of the semiconductor problem in the set of variables \( \psi, \Phi_p \) and \( \Phi_n \), the latter two being termed the Slotboom variables (cf. [11]). These are defined by the relationships

\[ p = n_{\text{int}} e^{\phi_s / \mu_r} \Phi_p \]  
\[ n = n_{\text{int}} e^{\phi_s / \mu_r} \Phi_n \]

In this case, equations (5)-(6) read

\[ J_p = -q\mu_r U_r n_{\text{int}} e^{-\phi / \mu_r} \nabla \Phi_p \]  
\[ J_n = q\mu_r U_r n_{\text{int}} e^{-\phi / \mu_r} \nabla \Phi_n \]

2. SINGULARLY PERTURBED CHARACTER

For the discussion in this paper, it will suffice to remark that, because of the behaviour of the electric field, the equations (7) and (8) can be considered as singularly perturbed first order differential equations for \( \psi \) and \( \phi \), respectively. This is an important observation, since it explains why standard difference schemes cannot be applied to the discretisation of these equations.

3. EXPONENTIAL FITTING

As has been discussed in the foregoing section, (5) and (6) can be considered as singularly perturbed first order differential equations for \( p \) and \( n \), respectively. Application of the standard central difference scheme on a mesh \( \{ x_0, \ldots, x_N \} \) leads to the following expression for the hole current density in the interval \( [x_i, x_{i+1}] \) (denoted by \( J_{p,i+1/2} \)):
where
\[ E_{i+1/2} = \frac{\psi_{i+1} - \psi_i}{x_{i+1} - x_i} \]
A similar expression is obtained for the electron current density. These expressions are then used in the discrete versions of equations \((1)-(3)\) for the latter can be carried over directly to the semiconductor.

This scheme has first been described by Scharfetter and Gummel (cf. \[1\]).

It is wellknown that this scheme yields non-stable solutions. In fact, for \(\tau\) taken close to 1, the discrete scheme does not guarantee existence of a solution in terms of the variables \(\psi_i, \Phi_{p,i}, \Phi_{n,i}\). Thus, the discrete scheme does not guarantee existence of a solution.

Using the relations between the carrier concentrations and the electric field \(E\) and the current densities \(J\) and \(\mathbf{J}\), we obtain:

\[ J_{p,i+1/2} = -q\mu_p (U_T \frac{P_{i+1}-P_i}{x_{i+1} - x_i} - \frac{P_{i+1} + P_i}{2} \frac{E_{i+1/2}}{e}) \]
\[ J_{n,i+1/2} = q\mu_n (U_T \frac{N_{i+1}-N_i}{x_{i+1} - x_i} - \frac{N_{i+1} + N_i}{2} \frac{E_{i+1/2}}{e}) \]
\[ \mathbf{J}_{p,i+1/2} = -q\mu_p U_{T,\text{int}} \frac{N_{i+1}-P_{i+1}}{e} \mathbf{E}_{i+1/2} - \frac{P_{i+1} + P_i}{2} \mathbf{E}_{i+1/2} \]
\[ \mathbf{J}_{n,i+1/2} = q\mu_n U_{T,\text{int}} \frac{P_{i+1}-N_{i+1}}{e} \mathbf{E}_{i+1/2} - \frac{N_{i+1} + N_i}{2} \mathbf{E}_{i+1/2} \]
\[ \mathbf{J}_{\text{p},i+1/2} = -q\mu_p U_{T,\text{int}} \frac{\psi_{i+1} - \psi_i}{x_{i+1} - x_i} \mathbf{E}_{i+1/2} - \frac{\psi_{i+1} + \psi_i}{2} \mathbf{E}_{i+1/2} \]
\[ \mathbf{J}_{\text{n},i+1/2} = q\mu_n U_{T,\text{int}} \frac{\psi_i - \psi_{i-1}}{x_{i+1} - x_i} \mathbf{E}_{i+1/2} - \frac{\psi_i + \psi_{i-1}}{2} \mathbf{E}_{i+1/2} \]

Rearranging this expression leads to the following expression for \(J_p\) in the interval \([x_i, x_{i+1}]\), which we again denote by \(J_{p,i+1/2}\):

\[ J_{p,i+1/2} = -q\mu_p U_{T,\text{int}} \psi_i \mathbf{E}_{i+1/2} \]
\[ J_{n,i+1/2} = q\mu_n U_{T,\text{int}} \psi_i \mathbf{E}_{i+1/2} \]
\[ \psi_i = \mathbf{E}_{i+1/2} \frac{P_{i+1} + P_i}{2} \]

Using the relations between the carrier concentrations and the electric field, we obtain:

\[ \psi_i = \mathbf{E}_{i+1/2} \frac{P_{i+1} + P_i}{2} \]

4. EXTENSION TO HIGHER DIMENSIONS

The box method is ideally suited for the discretisation of equations of the form

\[ \nabla \cdot \mathbf{F} = f \]

The starut point is a mesh consisting of triangles, rectangles, quadrilaterals, or other polygons. A so-called box \(B_i\) is constructed around each mesh point \(x_i\), in such a way that the union of all boxes is the entire simulation domain and that boxes do not overlap. The most common way of doing this is to construct \(B_i\) using the midpoints of the mesh sides. Having completed this construction, we integrate (21) over each of the boxes:

\[ \int_{B_i} \nabla \cdot \mathbf{F} dV = \int_{B_i} f dV \]

The discretisation is now completed by approximating the integrals in this equation. A standard procedure is to use the lowest order quadrature rule for both integrals. Thus, the right-hand side of (22) is approximated by \(\text{vol}(B_i) f_i\). The left-hand side is approximated by

\[ \sum_k l_k \mathbf{F} \cdot n_k \]

where the \(n_k\) are the normals in the midpoints of the mesh sides and \(l_k\) are the lengths of the corresponding box sides.

Equations (1)-(3) can be discretised using the box method. The remaining problem is to evaluate the normal components of the electric field \(\mathbf{E}\) and the current densities \(J_p\) and \(J_n\). Because of the construction of the boxes, however, this is rather straightforward. Namely, the normal components are exactly the components along the mesh sides. Thus, to obtain the desired quantities, we can just use the techniques outlined in the previous section. In other words: the one-dimensional arguments used to obtain exponentially fitted discretisations of the first order differential equations for \(J_p, J_n\) in between two mesh points can be used here. For the electric field the situation is even simpler, because we do not need any fitting. In that case we just use expressions of the form (10).
Conclusion:
The box scheme for equations (1)-(3), combined with the exponentially fitted schemes for equations (4)-(6), provide a suitable discretisation method for the semiconductor equations in 1-d, 2-d and 3-d.

In fact, we conjecture here that the proposed exponentially fitted box schemes yield uniform error estimates (this has been shown in 1-d and in the 2-d rectangular case).

5. EXPONENTIAL FITTING FOR SMALL SYSTEMS: THE CASE OF AVALANCHE GENERATION

In this section we generalise the Scharfetter-Gummel method to the case where the effect of avalanche generation is taken into account, using exponential fitting techniques for singularly perturbed 2 x 2 systems as described in [5]. This effect, which may lead to breakdown of the device, is accounted for in the equations by an extra recombination/generation term. Normally, these terms only depend on the carrier concentrations, not on the current densities. Therefore, (2) and (3) are first order equations in \( J_p \) and \( J_n \) without any zero order terms. From a mathematical point of view this means that the current densities will vary rather smoothly. Remark that the assumption of piecewise constant current densities made in the derivation of the Scharfetter-Gummel scheme agrees with this.

When avalanche generation is taken into account, the situation changes completely. In this case, an extra term has to be included of the form

\[
R_{12} = \frac{1}{q} (\alpha_p J_p + \alpha_n J_n)
\]

Now we see that the first order differential equations do contain a zero order term, thus allowing the possibility of rapidly (namely, exponentially) varying current densities. It is clear that the assumption of piecewise linear current densities may not be adequate in this case.

As a starting point for our derivation we take the homogeneous form of equations (2) and (3), i.e. \( R \) is equal to the impact ionisation term given in (23). The equations read

\[
J'_p = + \alpha_p J_p + \alpha_n J_n
\]

\[
J'_n = - \alpha_p J_p - \alpha_n J_n
\]

which can be written as

\[
J' = \lambda J
\]

where \( J = (J_p, J_n)^T \) and

\[
\lambda = \left( \begin{array}{cc} \nu_p \alpha_p & \nu_n \alpha_n \\ -\nu_p \alpha_p & -\nu_n \alpha_n \end{array} \right)
\]

Here, \( \nu_p = \text{sign}(J_p) \), \( \nu_n = \text{sign}(J_n) \) are the 'directions' of \( J_p \) and \( J_n \). Using equations (9)-(11) we see that \( \nu_p = -\text{sign}(\phi'_p) \), \( \nu_n = -\text{sign}(\phi'_n) \). Thus, (26) may be considered as a linear equation in \( J \).

Now we are ready to discretise the problem. As usual, we take a mesh \( \{x_0, \ldots, x_j\} \), the midpoint of the interval \([x_i, x_{i+1}]\) being denoted by \( x_{i+1/2} \). We assume that \( A \) is constant on each of these intervals (notation: \( A_{i+1/2} \)). Multiplying (26) on the interval \([x_i, x_{i+1}]\) by \( e^{-(x-x_{i+1})/A_{i+1/2}} \) we obtain:

\[
d(e^{-(x-x_{i+1/2})A_{i+1/2}}) = 0
\]

Thus, we may conclude that for solutions \( J \) of the continuous problem (26) we have that \( e^{-(x-x_{i+1/2})A_{i+1/2}} = \text{constant vector} \). To proceed we now use equations (13) and (14). These can be written in the form

\[
J = D \Phi'
\]

where \( \Phi = (\phi_p, \phi_n)^T \) and

\[
D = \left( \begin{array}{cc} -\mu_p U_T \text{Int} & 0 \\ 0 & -\mu_n U_T \text{Int} \end{array} \right)
\]

Since \( e^{-(x-x_{i+1})A_{i+1/2}} \) is a constant vector, which we will denote by \( J_{i-1/2} \) (substitute \( x = x_{i+1/2} \)) we have that

\[
\Phi' = D^{-1} e^{-(x-x_{i+1})A_{i+1/2}} J_{i+1/2}
\]

and it follows that

\[
\Phi_{i+1} - \Phi_i = \left( \int_{x_i}^{x_{i+1}} -1 A_{i+1/2} dy \right) J_{i+1/2}
\]

From this we obtain a discrete expression for the current densities on the interval \([x_i, x_{i+1}]\):

\[
J(x) = e^{-(x-x_{i+1})A_{i+1/2}} \int_{x_i}^{x_{i+1}} J_{i+1/2} dy
\]

Now we see that the assumption of piecewise constant current densities made in the derivation of the Scharfetter-Gummel scheme agrees with this.

The final discretisation of (26) is then obtained by integrating over the box \([x_{i-1/2}, x_{i+1/2}]\):

\[
J_{i+1/2} - J_{i-1/2} = \int_{x_{i-1/2}}^{x_{i+1/2}} A_{i+1/2} J_i(x) dx
\]

Equation (29) can be expanded further by substituting the expressions (27) into the integrals. This leads to the following discretisation.

\[
\left( e^{-(x-x_{i+1})A_{i+1/2}} \right) J_{i+1/2} - \left( e^{-(x-x_{i-1})A_{i-1/2}} \right) J_{i-1/2} = 0
\]

References


GRID APPROXIMATION OF BOUNDARY VALUE PROBLEMS FOR QUASILINEAR SINGULARLY PERTURBED ELLIPTIC EQUATIONS IN THE CASE OF FULL DEGENERATION

J.J.H. MILLER
School of Mathematics
Trinity College Dublin

and

GRIGORII I. SHISHIKIN
Institute of Mathematics and Mechanics
Ural Branch of USSR Academy of Sciences
Sverdlovsk, USSR

Abstract. A quasilinear singularly perturbed elliptic equation degenerating into an equation not containing derivatives is considered. Problems which appear when constructing numerical methods based, in particular, on fitting techniques are discussed. In the case of the Dirichlet problem on a strip, or on a domain with smooth curvilinear boundary, principles for the construction of uniformly (with respect to the parameter) convergent schemes are considered.

I. PROBLEM FORMULATION

On a strip \( D = \{ x : 0 < x_1 < d, |x_2| < \infty, s = 2, \ldots, n \} \) the following Dirichlet problem for a quasilinear equation is considered

\[
L(u(x)) \equiv \varepsilon^2 L'(u(x))u(x) - f(x, u(x)) = 0, \quad x \in D, \quad u(x) = \varphi(x), \quad x \in \Gamma.
\]

Here \( L'(u) \) is the second order elliptic operator

\[
L'(u) \equiv \sum_{k=1}^{n} a_{sk}(x, u) \frac{\partial^2}{\partial x_s \partial x_k} + \sum_{k=1}^{n} b_k(x, u) \frac{\partial}{\partial x_s} - c(x, u),
\]

It is assumed that the coefficients of the operator \( L' \) and the functions \( f \) and \( \varphi \) are sufficiently smooth, and that for the function \( f(x, u) \) the following condition is satisfied

\[
(\partial/\partial u)f(x, u) \geq \alpha > 0, \quad (x, u) \in \overline{D} \times R^1.
\]

The parameter \( \varepsilon \) can take any value in \( (0, 1] \). When \( \varepsilon \) tends to zero in a neighbourhood of the boundary \( \Gamma \) a boundary layer appears. This boundary layer is described by an ordinary differential equation. Derivatives of the solution along all directions out of a neighbourhood of the boundary layer, and also derivatives along directions collinear with the boundary domain in the neighbourhood of the boundary layer are bounded uniformly with respect to the parameter. The numerical solution of such boundary value problems is difficult even for linear equations and gives rise to the problem of constructing approximations on special grids, which converge uniformly with respect to the parameter (see, for example, [1]).

II. TYPICAL NUMERICAL PROBLEMS FOR NONLINEAR EQUATIONS

On a uniform grid it is impossible to construct a numerical solution of problem (1) converging uniformly with respect to the parameter. This is because, on the one hand problem (1) is nonlinear and, on the other hand the difference of the values of the solution to problem (1) on neighboring nodes of the grid does not converge to zero uniformly with respect to the parameter (when the grid size tends to zero). For this reason it is necessary to construct difference schemes on grids condensing in the boundary layer.

III. THE BOUNDARY VALUE PROBLEM ON THE STRIP

First suppose that the coefficients of the operator \( L' \) do not depend on \( u(x) \). Then the solution to problem (1) is the limit of a sequence of solutions to the boundary value problem

\[
L^2 u^{(k)}(x) \equiv \varepsilon^2 L'(u^{(k)}(x)) = f_\alpha(x, u^{(k-1)}(x)), \quad x \in D, \quad u^{(k)}(x) = \varphi(x), \quad x \in \Gamma
\]

Note that this equation is linear with respect to \( u^{(k)}(x) \) and that \( f_\alpha(x, u) = f(x, u) - \alpha u \). To solve problem (1) the iterative difference scheme

\[
L^2 z^{(k)}(x) = f(x, z^{(k-1)}(x)), \quad x \in D_h, \quad z^{(k)}(x) = \varphi(x), \quad x \in \Gamma_h
\]

approximating problem (2) is used. Derivatives are approximated by classical finite differences on rectangular grids (see, for example, [2]), which are condensed in a neighborhood of the boundary layer by a special rule (one such grid condensing rule is given in [3]). The conditions which guarantee uniform with respect to the parameter convergence of the solution to the solution of problem (3) (when \( k \to \infty \) and the number of nodes increases) are satisfied. A similar method is used when the coefficients of the operator \( L' \) depend on the solution of problem (1).

IV. DOMAINS WITH CURVILINEAR BOUNDARIES

In the case of domains with smooth curvilinear boundary values iterative schemes based on the alternating method of Schwartz are constructed. A neighborhood of the boundary \( \Gamma \) is covered by a system of overlapping subdomains. In each subdomain the boundary \( \Gamma \) of the
initial domain \( D \) is rectified by using a coordinate transformation. A difference scheme (in the new coordinate system) is constructed by employing results obtained for the boundary value problem on the strip. For boundary value problems on domains with curvilinear boundaries, conditions guaranteeing uniform with respect to the parameter convergence of the constructed difference scheme are deduced.

REFERENCES


SPLITTING-TIME AND EXPONENTIAL FITTING-SPACE DISCRETIZATIONS FOR CONVECTION-DIFFUSION PROBLEMS.

C. CLAVERO, J.C. JORGE, F. LISBONA.
Departamento de Matemática Aplicada,
Universidad de Zaragoza,
Zaragoza (Spain).

Abstract. In this paper we develop a finite-difference method for space dependent convection dominated convection-diffusion problems, modeled by the singularly perturbed parabolic equation
\[ \frac{\partial u}{\partial t} - \varepsilon \Delta u + \nabla \cdot \mathbf{w} u + ku = f, \]
in \( \Omega \times [0,T] \),
where \( \Omega \subset \mathbb{R}^2 \), \( k \geq 0 \) and \( \varepsilon \) is a (possibly small) positive constant.

This method may be viewed as a combination between Fractional Steps time semi-discretizations, and one-dimensional space discretizations of exponential fitting type. Making use of the consistency and the contractivity of the time integration process, and the special properties of the discretizations made, via exponential fitting, for the space differential operators at each time level, a (uniform in \( \varepsilon \)) convergence result is proven. Important advantages for computations are also obtained.

I. INTRODUCTION

Let \( \Omega \) be a bounded domain in \( \mathbb{R}^2 \) with a piecewise smooth boundary \( \Gamma \). We will deal with the numerical approximation of the solution of time dependent, convection dominated, convection-diffusion problems defined by the equation
\[ \frac{\partial u}{\partial t} - \varepsilon \Delta u + \nabla \cdot \mathbf{w} u + ku = f, \]
in \( \Omega \times [0,T] \),
and the initial-boundary conditions
\[ u(x,0) = u_0(x) \text{ in } \Omega \]
\[ u(x,t) = g(x,t) \text{ in } \Gamma \times [0,T], \]
where \( \mathbf{w} = (w_1, w_2) \) and \( k \) are smooth functions on \( \Omega \), with \( \varepsilon > 0 \) and such that it may occur \( \varepsilon \ll |w| \). In general, the solution of this problem is not globally smooth (even for smooth data) but it may present rapid variations in certain narrow regions in \( \Omega \) (layers).

Problems of this kind are found in the modelling of convection-conduction of heat and atmospheric transport of pollutants. It is well known that standard finite difference or finite element methods applied to convection dominated flow problems give unphysical oscillatory solutions with a reasonable mesh size; but it may present rapid variations in certain narrow regions in \( \Omega \) (layers).

II. TIME SEMIDISCRETIZATION

For simplicity in the exposition of our analysis we will consider the following initial Dirichlet boundary value problem
\[ \frac{\partial u}{\partial t} - \varepsilon \Delta u + \nabla \cdot \mathbf{w} u + ku = f, \]
in \( \Omega \times [0,1] \times [0,1] \times [0,T] \)
where \( k(x,y,t) \geq 0 \) and \( \mathbf{w} = (w_1(x,y,t), w_2(x,y,t)) \) such that \( w_i \) has constant sign, \( i = 1,2 \). We will assume compatibility between the initial and the boundary conditions, and \( k, \mathbf{w}, u, f \) smooth enough, to guarantee that the solution \( u \) of (1) verifies
\[ (u, A_1 u, A_2 u, A_1^2 u, A_2^2 u, A_1 A_2 u, A_2 A_1 u) \subset C^2(\Omega), \]
where
\[ A_1 \equiv -\varepsilon \frac{\partial^2}{\partial x^2} + \frac{\partial}{\partial x} + k_1, \]
\[ A_2 \equiv -\varepsilon \frac{\partial^2}{\partial y^2} + \frac{\partial}{\partial y} + k_2, \]
and \( k_1 \geq 0 \) and \( k_1 + k_2 = k \).

Note that if \( f \) is smooth, conditions (2) guarantee \( (u, A_1 u, A_2 u, A_1^2 u, A_2^2 u, A_1 A_2 u, A_2 A_1 u) \subset C^2(\Omega) \).

Let \( \gamma \) be the space boundary of \( \Omega \).

Problem (1) is discretized in time, so that we obtain semi-discrete approximations \( u^n(x,y) \) to \( u(x,y,t_n) \) solution of (1) at the instant of time \( t_n = n \Delta t \) by means of the following Fractional Steps Scheme
\[ \left\{ \begin{array}{ll}
\text{a) } u^0 = u_0(x,y); \\
\text{b) } u^{n+1}(0,y) = u^n(y, t_n + 1), \\
\text{c) } u^{n+1}(1,y) = u^n(y, t_n + 1); \\
\text{d) } u^{n+1}(x,0) = u^n(x, t_n + 1), \\
\text{e) } u^{n+1}(x,1) = u^n(x, t_n + 1). \\
\end{array} \right. \]

Note that if we define \( "(I + \Delta A_1)^{-1} u" \) by the resolution of the stationary problem
\[ \left\{ \begin{array}{ll}
(I + \Delta A_1) x = u, \\
x(0,y) = 0, \\
x(1,y) = 0; \\
\end{array} \right. \]
and analogously for \( "(I + \Delta A_2)^{-1} u" \), we have that the linear operators \( "(I + \Delta A_1)^{-1} u" \) are inverse monotonous and
\[ \| (I + \Delta A_1)^{-1} u \| \leq 1, i = 1,2. \]
We define the Local Error for (3)

\[ e_n = u(t_{n+1}) - \hat{u}^{n+1} \]  

where \( \hat{u}^{n+1} \) is the result \textit{"u}^{n+1} \textit{"} obtained of applying scheme (3) taking \( u^n = u(t_n) \).

Under the hypothesis (2) for the solution \( u \) of the continuous problem (1), and assuming \( f_1 \) and \( f_2 \) are smooth, we obtain

\[ \|e_n\|_\infty \leq C(\Delta t^2) \quad \text{(consistency of the scheme (3),)} \]

and taking into account (6) and (4) is easy to prove

\[ \|u(t_n) - u^n\|_\infty \leq C(\Delta t \text{ (convergence of the semidiscretization).}) \]

II. TOTAL DISCRETIZATION

Description: In order to obtain a total discrete method to approach (1), scheme (3) is discretized in space using an exponential fitting technique.

Let \( \Omega_h \) be a rectangular grid (not necessarily uniform) of points \( (x, y) \in [0, 1] \times [0, 1] \) and let us denote \([j, k]\), the restriction of a function defined in \([0, 1] \times [0, 1] \) to \( \Omega_h \). Our total discretization obtains approximations \( u_h \) to \( u \) by means of the following algorithm

\[ T_{1,h,k}u_h(x, y) = u_h(x, y) + \Delta t f_1(x, y) \]

where \( T_{1,h,k} \) and \( \Delta t \) are the result of discretizing, via exponential fitting, the uniparametric families of one-dimensional singularly perturbed elliptic problems (3b) and (3c) respectively.

For the construction of \( T_{1,h,k} \) and similarly for \( T_{2,h,k} \) on each time level \( t \) we proceed in the following way:

Let \( \Omega_h \) be defined as the set of functions \( \{u(x, y) | u(x, y) \in \Omega_h \} \subset \Omega_h \) with \( \theta = x_0 < x_1 < \ldots < x_N = 1 \), and \( \min_{i=1}^N u(x_i, y) \).

On \( \Omega_h \) we define the exponential fitting scheme in the form

\[ T_{1,h,k}u_h = Q_{1,h,k}T_{r,h,k} u_h, \]

where\[ Q_{1,h,k} \] and \( T_{r,h,k} \) are the result of discretizing, via exponential fitting, the uniparametric families of one-dimensional singularly perturbed elliptic problems (3b) and (3c) respectively.

Finally we will remark that, in order to obtain a good speed-up, the algorithm can be easily implemented on parallel computers, since the resolution at each time level involves a set of uncoupled linear systems.

Numerical experiments will be presented in the oral session and the complete version of this paper.
Abstract. A model singularly perturbed boundary value problem with a single shock layer is considered. Various numerical methods are analyzed and difficulties in resolving the layer are discussed.

The Continuous Problem

We consider the following singularly perturbed boundary value problem:

\[ Tu := -\varepsilon u'' - uu' + c(x)u = 0, \quad x \in [0, 1], \quad (1) \]

\[ Bu := (u(0), u(1)) = (U_0, U_1), \quad (2) \]

where \( 0 < \varepsilon \ll 1 \), \( U_0 \) and \( U_1 \) are given numbers, \( \varepsilon \) is a sufficiently smooth function and

\[ c(x) \geq c > 0, \quad x \in I. \]

Thus, the famous Lagerstrom-Cole model problem is included. The problem (1-2) has a unique solution \( u \). We assume that \( U_0 < 0 < U_1 \), and thus \( u(x) > 0, \quad x \in I \). Furthermore, let

\[ u_i = \int_i^x c(t)dt + U_i, \quad i = 0, 1, \quad (3) \]

be the solutions to the reduced problem (1) with \( \varepsilon = 0 \), and let there exist a point \( x^* \in (0, 1) \) such that \( u_0(x^*) + u_1(x^*) = 0 \). Then \( u \) has a shock layer at \( x = x^* \), cf. [1]. By using the technique from [5] we can estimate the derivatives of \( u \):

\[ |u'(x)| \leq M[1 + \varepsilon^{-k} \exp(-m|x-x^*|/\varepsilon)], \quad x \in I. \quad (4) \]

Here \( x^* \) is the unique point in \( (0, 1) \) such that \( u(x^*) = 0, \quad m > 0 \) is an arbitrary constant independent of \( \varepsilon \), and by \( M \) we denote throughout a positive generic constant independent of \( \varepsilon \). \( x^* \) is not known usually, but it can be approximated well by \( x^* \). Moreover, \( u_i, i = 0, 1 \), are good approximations to \( u \) outside the layer:

\[ |u_i(x) - u(x)| \leq M[\varepsilon + \exp(-m|x-x^*|/\varepsilon)], \quad x \in I_i, \quad i = 0, 1, \quad (5) \]

where \( I_0 = [0, x^*] \) and \( I_1 = [x^*, 1] \). This can be proved by using inverse monotonicity of the linear operator \( L_0 = \varepsilon^{1/2}u'' \) with appropriate boundary operators. For \( x \in I_i \) it holds that

\[ LM[|x\exp(-\varepsilon^{-1}\int_{x_1}^x u(t)dt)|] \geq \pm 2u_i(x) = \pm L[u(x) - u_i(x)], \quad (6) \]

and then (5) follows by the technique from [5].

Numerical Methods

The standard numerical method for solving (1-2) is discretization by the Engquist-Osher or some similar finite-difference scheme, [4]. We are interested in resolving the layer accurately, and because of (4) with \( x^\circ \) approximated by \( x^* \) it seems reasonable to apply a special discretization mesh which is dense near \( x^* \). This approach gives good results in the case of boundary layers. [4,6] - the pointwise errors are first order accurate uniformly in \( \varepsilon \). Here, however, the uniform accuracy is present in the discrete \( L^1 \) norm only, and the use of the special mesh is not justified. The numerical values cluster around a point different from \( x^* \). Another possibility is to apply an iterative mesh construction, such as the method from [7]. Neither this gives satisfactory results. The numerical solution looks fine qualitatively, but the layer is shifted from \( x^* \). There is only one explanation for these phenomena: the discrete problem has its own layer which is shifted from the continuous one.

The best, but not ideal, results can be obtained by the technique which was applied in [3] to a boundary layer quasilinear problem. Let \( k = -a/\varepsilon \) with a positive constant \( a \), and let \( s_0 = x^* - k\varepsilon, \quad s_1 = x^* + k\varepsilon \). Then from (6) it follows that

\[ |u(x) - u_i(x)| \leq ME \]

for \( x \in [0, s_0) \) if \( i = 0 \), and \( x \in (s_1, 1] \) if \( i = 1 \). Thus we shall approximate \( u \) by \( u_0 \) and \( u_1 \), in corresponding intervals \([u_i(x)] \), can be obtained from (3), either exactly or numerically, and it remains to solve the problem

\[ Tu = 0, \quad x \in [s_0, s_1], \quad u(x) = u_i(x), \quad i = 0, 1, \quad (6) \]

which is practically unperturbed. We shall do this by applying the standard central scheme on the equidistant mesh with the step \( h = 2\varepsilon/n \), \( n \in \mathbb{N} \). The scheme is stable in the discrete \( L^1 \) norm if \( n \) is sufficiently great, independently of \( \varepsilon \). An estimate of the error of this procedure can be obtained in \( L^1 \) norm since the operator \((T, B)\) is \( L^1 \)-stable. However, this is not sharp enough for we are interested in pointwise accuracy. The pointwise errors will be illustrated by the example with \( c(x) = 1 \) and \( U_0 = -1, \quad U_1 = 1/2 \). Only the errors of the numerical solution of (6) are interesting. The numerical solution is compared to the inner solution of the problem, [1]. For \( \alpha = 5 \) and \( \varepsilon = 10^{-4} \) we get the errors 8.50E-2, 1.51E-2, 4.40E-3, 4.50E-3 for \( n = 10, 20, 40, 50 \), respectively. The errors for the same \( \alpha \) and \( n \) but for \( \varepsilon = 10^{-4} \) are 1.35E-1, 4.13E-2, 7.65E-3, 1.71E-3.

References

AN ARBITRARY ORDER DIFFERENCE SCHEME FOR A NONSELF-ADJOINT SINGULAR PERTURBATION PROBLEM IN CONSERVATION FORM

Pengcheng Lin
Department of Computer Science
Fuzhou University
Fuzhou, 350002, PR of China

Jiansheng Zheng
Xiamen International Book Centre.
Xiamen, 361004, PR of China

Abstract A nonself-adjoint singular perturbation problem in conservation form is considered. We construct a three point difference scheme and prove that the solution of the difference scheme converges uniformly in small parameter $\varepsilon$ with order $h^{n+1}$ to the solution of the differential problem.

I. INTRODUCTION

A nonself-adjoint singular perturbation problem in conservation form

\[ \begin{aligned}
&\alpha(x) u'' + f(x) u' + g(x) u = h(x), 0 < x < 1 \\
&u(0) = \mu_0, u(1) = \mu_1 
\end{aligned} \]

is considered, where $\varepsilon$ is a parameter in $(0,1], g(x), f(x), h(x)$ are given functions, $\mu_0, \mu_1$ are given constants, $m$ is a positive integer which is given arbitrarily. Under these conditions, there is an unique solution for the equation (1).

Recently, some authors derived an accurate difference scheme of order one or two for (1). In this paper, we construct a three-point difference scheme for (1). For any $m > 0$, we prove that the solution of the difference scheme converges uniformly in $\varepsilon$ with order $h^{n+1}$ to the solution of (1).

II. THE DIFFERENCE SCHEME

A. An accurate difference scheme

We construct an accurate difference scheme for the problem (1)

\[ \begin{aligned}
L_i u_i &= v L_i (0) u_{i+1} - u_i + v(0) u_{i+1} = -v\phi_i(0), \\
&\quad i = 1, 2, \ldots, N - 1 \\
&\quad u_0 = \mu_0, u_N = \mu_1,
\end{aligned} \]

where $\phi_i (k = 0, 1, 0)$ satisfies:

\[ \begin{aligned}
&v_i(x, \varepsilon) = p(x) v_i(x, \varepsilon) + \frac{d}{dx} v_i(x, \varepsilon) \equiv v_i(x, \varepsilon) v(x, \varepsilon) \\
&v_0(x) = v(0), v_{N+1}(x) = 0
\end{aligned} \]

B. Approximate difference scheme

We establish the following approximate difference scheme.

\[ \begin{aligned}
&\frac{\alpha(x) v' + f(x) v + g(x) v}{\varepsilon} = h(x), 0 < x < 1 \\
&v(0) = 0, v(1) = 1
\end{aligned} \]

and

\[ \begin{aligned}
&\frac{\alpha(x) w' + f(x) w + g(x) w}{\varepsilon} = h(x), 0 < x < 1 \\
&w(0) = 0, w(1) = 1
\end{aligned} \]

the exact solution is

\[ u(x) = c_1 \sqrt{1 + x} + c_2 (1 + x)^{-\frac{1}{2}} + \frac{1 + x}{1 + \varepsilon} \]

C. The three-point difference scheme

We construct the three-point difference scheme for (1), such as:

\[ \begin{aligned}
&\frac{\alpha(x) v' + f(x) v + g(x) v}{\varepsilon} = h(x), 0 < x < 1 \\
&v(0) = 0, v(1) = 1, v_{N+1} = 0
\end{aligned} \]

III. THE CONVERGENCE THEOREM

We will obtain the following uniform convergence result.

Theorem Suppose $u(x)$ be the solution of the difference scheme (6) and $u(x)$ be the solution of (1), then we have

\[ u(x) = u(x) \leq M h^{n+1}, \]

where $A$ is a constant independent of $\varepsilon$ and $h$.

IV. NUMERICAL EXAMPLE

Consider the following problem

\[ \begin{aligned}
&\frac{\alpha(x) v' + f(x) v + g(x) v}{\varepsilon} = h(x), 0 < x < 1, \\
&v(0) = 0, v(1) = 1
\end{aligned} \]

(7)

This work supported in part by a grant of the National Science Foundation of China and in part by a grant of the Natural Science Foundation of Fujian.
where 
\[ c_1 = \frac{(2 - \frac{1}{2} - 1 + \varepsilon)}{(1 + \varepsilon)(\sqrt{2} - 2 - \frac{1}{2})}, \]
\[ c_2 = \frac{(1 - \sqrt{2} - \varepsilon)}{(1 + \varepsilon)(\sqrt{2} - 2 - \frac{1}{2})}, \]
\[ E_m = \max_{i} |u_i^j - u(x_i)|, \]
where \( u(x_i) \) is the solution of (7), \( u_i^j \) is the solution of the scheme (6) when \( m = 1 \). Numerical result lists the following table.

<table>
<thead>
<tr>
<th>( h = 0.1 )</th>
<th>( j )</th>
<th>1</th>
<th>6</th>
<th>9</th>
<th>( E_m )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \varepsilon = 1 )</td>
<td>( u(x_i) )</td>
<td>0.1705956</td>
<td>0.6971326</td>
<td>0.9275933</td>
<td>5.286932 \times 10^{-5}</td>
</tr>
<tr>
<td>( \varepsilon = 1 )</td>
<td>( u_i^j )</td>
<td>0.1705048</td>
<td>0.6971323</td>
<td>0.9275981</td>
<td>1.7007319 \times 10^{-4}</td>
</tr>
<tr>
<td>( \varepsilon = 1 )</td>
<td>( u(x_i) )</td>
<td>0.3551245</td>
<td>0.7113618</td>
<td>0.9296028</td>
<td>1.850609 \times 10^{-4}</td>
</tr>
<tr>
<td>( \varepsilon = 0.1 )</td>
<td>( u(x_i) )</td>
<td>0.2659307</td>
<td>0.7201022</td>
<td>0.9295832</td>
<td>1.7007319 \times 10^{-4}</td>
</tr>
<tr>
<td>( \varepsilon = 0.1 )</td>
<td>( u_i^j )</td>
<td>0.2658968</td>
<td>0.7201026</td>
<td>0.9296028</td>
<td>1.7007319 \times 10^{-4}</td>
</tr>
<tr>
<td>( \varepsilon = 0.1 )</td>
<td>( u(x_i) )</td>
<td>0.3552783</td>
<td>0.7114487</td>
<td>0.9268539</td>
<td>1.7007319 \times 10^{-4}</td>
</tr>
<tr>
<td>( \varepsilon = 0.1 )</td>
<td>( u_i^j )</td>
<td>0.3552783</td>
<td>0.7114487</td>
<td>0.9268539</td>
<td>1.7007319 \times 10^{-4}</td>
</tr>
<tr>
<td>( \varepsilon = 0.025 )</td>
<td>( u(x_i) )</td>
<td>0.621421 \times 10^{-2}</td>
<td>0.7535943</td>
<td>0.9823888</td>
<td>5.745388 \times 10^{-5}</td>
</tr>
<tr>
<td>( \varepsilon = 0.025 )</td>
<td>( u_i^j )</td>
<td>0.620246 \times 10^{-2}</td>
<td>0.7535513</td>
<td>0.9823595</td>
<td>5.745388 \times 10^{-5}</td>
</tr>
<tr>
<td>( \varepsilon = 0.025 )</td>
<td>( u(x_i) )</td>
<td>0.3101401</td>
<td>0.7423608</td>
<td>0.9813186</td>
<td>7.778406 \times 10^{-6}</td>
</tr>
<tr>
<td>( \varepsilon = 0.025 )</td>
<td>( u_i^j )</td>
<td>0.3101479</td>
<td>0.7423628</td>
<td>0.9813186</td>
<td>7.778406 \times 10^{-6}</td>
</tr>
</tbody>
</table>

The numerical results show that the numerical experiment coincides with the theoretical analysis.

The authors gratefully acknowledge the support of K. C. Wong Education Foundation Hong Kong.

References


THE DIFFERENCE METHOD FOR SOLVING SINGULAR PERTURBATION PROBLEMS
OF THE PARABOLIC PARTIAL DIFFERENTIAL EQUATIONS
INVOLVING TWO SMALL PARAMETERS

Fengcheng Lin
Department of Computer Science
Fuzhou University
Fuzhou, 350002, PR. of China

Meifeng Yang
Fujian Branch
Cyts tours Corporation
Fuzhou, 350001, PR. of China

Abstract The singular perturbation problem of the parabolic partial differential equations involving two small parameters is considered. We construct an exponentially fitted difference scheme and prove that when $t \geq \delta t, \delta t = \frac{1}{x} \leq 1 - \delta x$ (where $\delta$ is a small positive number), the solution of the difference scheme converges uniformly to the solution of the original differential equation with order one.

1. INTRODUCTION

In this paper we consider the singular perturbation problem of the parabolic partial differential equations involving two small parameters. Wang Guoying [1] constructed a difference scheme with fitted factors. Applying the classical and non-classical estimation method to this scheme, he proved that the solution of this scheme converges uniformly to the original differential equation with $O(\delta t + \delta x)$. In this paper, using the method of decomposing the singular term from its solution and combining an asymptotic expansion of the equation, we prove that the solution of the difference scheme converges uniformly to the solution of the differential problem with $O(\delta t + \delta x)$.

II. CONTINUE PROBLEM

We consider the following problem

\[
\begin{align*}
L_{\mu}u &= -\frac{\partial u}{\partial t} + \epsilon \frac{\partial^2 u}{\partial x^2} - \alpha(x,t) \frac{\partial u}{\partial x} - b(x,t)u \\
&= f(x,t), \quad 0 < x < 1, 0 < t < T \\
u(0,t) &= u_0(t), \quad u(1,t) = u_1(t), \quad t \in (0,T)
\end{align*}
\]

where $Q = (x,t), 0 \leq x \leq 1, 0 \leq t < T$ and $\Gamma = \{(x,t), 0 \leq x \leq 1, t \in (0,\infty)\} \cup \{(1,t), 0 \leq t \leq T\}$. We assume:

(H1): $f, \alpha, b \in C^1(Q), u_0 \in C^1(0,1), u_1 \in C^1([0,1])$

\[
g_0(0) = u_0(0), \quad g_1(0) = u_1(0),
\]

where $Q = \{(x,t), 0 < x < 1, 0 < t < T\}$ and $\Gamma = \{(x,t), 0 < x < 1\} \cup \{(1,t), 0 < t < T\}$.

(H2): $f(x,t) > 0, b(x,t) > 0, \forall (x,t) \in (0,1) \times (0,T)$.

The parameters $\epsilon, \mu$ are positive small parameters. When $\epsilon \to 0, \mu \to 0$ (1) is degenerated to $-b(x,t)u = f(x,t)$, therefore it loses one boundary condition in each side $x = 0, x = 1, t = 0$, it appears boundary layer phenomenon.

Suppose problem (1) has the following form of the asymptotic solution

\[
u(x, t) = \sum_{n=0}^{\infty} \mu^{n} u_{\delta n}(x, t) + \nu_{\delta n}(x, t)
\]

where $u_{\delta n}$ be the solution of the degenerated problem, $\nu_{\delta n}(x, t)$ are the boundary layer function near $x = 0$ and $x = 1$.

By Vishik-Lyusternik asymptotic method we obtain (take only first order approximation and let $v = \nu, \omega = \omega_0, Z = Z_0$)

\[
\begin{align*}
&\nu(x, t) = \left[ u_0(t) - \alpha(x,t)u_1(t) \right] + \epsilon \left[ \frac{\partial u_0(t)}{\partial x} - \alpha(x,t) \frac{\partial u_1(t)}{\partial x} \right] \\
&+ \epsilon^2 \left[ \frac{\partial^2 u_0(t)}{\partial x^2} + \alpha(x,t) \frac{\partial^2 u_1(t)}{\partial x^2} \right] + \cdots
\end{align*}
\]

\[
\exp \left( \frac{-\alpha(x,t) \frac{\partial u_1(t)}{\partial x}}{2} \right)
\]

we obtain

\[
\begin{align*}
&\left| D^{(1)} v_0(\xi, t) \right| \leq C \epsilon^{-1} \exp \left( -\lambda_{10} \right) \\
&\left| D^{(1)} v_0(\xi, t) \right| \leq C, \quad (x, t) \in Q \\
&\left| D^{(1)} u_0(\xi, t) \right| \leq C \epsilon^{-1} \exp \left( -\lambda_{10} \right) \\
&\left| D^{(1)} u_0(\xi, t) \right| \leq C, \quad (x, t) \in Q \\
&\left| D^{(1)} Z_0(x, \eta) \right| \leq \epsilon^{-1} \exp \left( -\frac{\beta_1}{\mu} \right) \\
&\left| D^{(1)} Z_0(x, \eta) \right| \leq \epsilon^{-1} \exp \left( -\frac{\beta_1}{\mu} \right) \\
&\left| (x, t) \in Q - \nu_1
\end{align*}
\]

\[
\text{This work supported in part by a grant of the National Natural Science Foundation of China and in part by a grant of the National Science Foundation of Fujian.
}

518
III. DISCRETE PROBLEM

Divide region $Q$ into rectangular mesh, $x_i = ih, j = 0, 1, \ldots, J, h = \frac{T}{J}$. Define mesh space

$$
\mathcal{V}^{h,\omega} = \{v(x_i, t_j) \mid 0 \leq i \leq J, 0 \leq n \leq N\}
$$

Denote

$$
\mathcal{G}(x, t) = \left. \frac{G(x, t)}{\Delta t} \right|_{n}\quad G(x, t) = G(x, 0)
$$

where $g_j$ is the function defined in $Q^{2,\omega}$.

Take

$$
\mu_j = \frac{\Delta t b(x, 0)}{2} \exp \left\{ \frac{-b(x, 0) \Delta t}{\mu} \right\}
$$

$$
\sigma_1 = -\frac{h^2 b}{4h} \left(1 + \frac{h}{2} \frac{h}{2} + \frac{h}{2} \right),
$$

$$
\sigma_2 = \frac{hb}{2} \left(1 + \frac{h}{2} \frac{h}{2} + \frac{h}{2} \right),
$$

Define the difference operator

$$
L^{J,\mu} \omega^{n+1} = -\mu^{n+1} D + \omega^{n+1} + c \sigma_1 \omega^{n+1} - b(x, t) \omega^{n+1}.
$$

This operator satisfies the maximum principle.

We construct the difference scheme

$$
L^{J,\mu} u^{n+1} = f^{n+1} \quad \text{in } Q^{J,\mu},
$$

$$
\begin{align*}
0 < n < N - 1; \quad & u^n = g_0(x, t), \quad u^0 = g_1(x, t), \quad 0 < n < N - 1; \\
& u^{n+1} = u(x_i, t_j), \quad 1 < j < J - 1.
\end{align*}
$$

Define the mesh functions $v_j, \omega_j, Z_j, G_j$ in the following

| $L^{J,\mu} v_j = L \omega(x_i, t_j)$ |
| \(v_j = v(x_i, t_j)\) |
| \(v_0 = v(0, t_j)\), \(v_J = v(1, t_j)\) |
| $L^{J,\mu} \omega_j = L \omega(x_i, t_j)$ |
| \(\omega_j = \omega(x_i, t_j)\) |
| \(\omega_0 = \omega(0, t_j), \quad \omega_J = \omega(1, t_j)\) |
| $L^{J,\mu} Z_j = L Z(x_i, t_j)$ |
| \(Z_j = Z(x_i, t_j)\) |
| \(Z_0 = Z(0, t_j), \quad Z_J = Z(1, t_j)\) |
| $L^{J,\mu} G_j = L G(x_i, t_j)$ |
| \(G_j = G(x_i, t_j)\) |
| \(G_0 = G(0, t_j), \quad G_J = G(1, t_j)\) |

then $u^{n+1} = v^{n+1} + \omega^{n+1} + Z^{n+1} + G^{n+1}$.

IV. MAIN THEOREM

Theorem 1 If $u(x, t) = v(x, t) + \omega(x, t) + Z(x, t) + G(x, t)$

$$
\begin{align*}
\text{then when } t & \geq M \mu^{n+1}, \quad M^2 \leq x \leq 1, \quad M^2, \quad (0 < \sigma < 1) \\
(1) & \text{then when } t \geq M \mu^{n+1}, \quad M^2 \leq x \leq 1, \quad M^2, \quad (0 < \sigma < 1)
\end{align*}
$$

\begin{align*}
& \left\{ \begin{array}{l}
G(x, t) = M(h + \Delta t), \\
|v(x, t) - v| \leq M(h + \Delta t), \\
|\omega(x, t) - \omega| \leq M(h + \Delta t), \\
|Z(x, t) - Z| \leq M(h + \Delta t), \\
|G(x, t) - G| \leq M(h + \Delta t),
\end{array} \right.
\end{align*}

Theorem 2 Assume that the coefficient and the right hand, initial, boundary functions of (1) are sufficiently smooth, and satisfy conditions (H1) and (H2), then when $t \geq M \mu^{n+1}, \quad M^2 \leq x \leq 1 - M^2$, the solution of the difference scheme (2) converges uniformly to the solution of (1), as $h \to 0$, $\Delta t \to 0$, and the following estimation holds.

$$
|u^n - u(x_i, t_j)| \leq M(\delta (h + \Delta t)),
$$

where $\delta$ is an arbitrary number in $(0, 1)$.

The authors gratefully acknowledge the support of K. C. Wong education foundation HongKong.

References


Abstract: A kind of nonlinear difference equations is considered. Singular perturbation method is applied to construct the asymptotic approximation of the solution to the difference equation. Using the theory of exponential dichotomies we show that the solution of an order-reduced equation is a good approximation of the solution to the difference equation except near boundaries. The corrections, which yield asymptotic approximations, are constructed.

1. INTRODUCTION

We are considering singularly perturbed difference-boundary problems of the form

\[ \eta_k = \alpha \eta_{k+1} + g(u_{k+1}) - g(u_k) - f(u_k), \quad k = 1, \ldots, n, \quad 0 < \epsilon \leq \epsilon_0 \quad \text{and} \quad u_0 = \alpha, \quad u_{n+1} = \beta, \]

They seem the "upwind" difference approximation of the following class of boundary value problems for nonlinear, second order, ordinary differential equations

\[ \epsilon g(u)u_+ + f(u)_+ + \beta(x)x = 0, \quad u(0) = \alpha, \quad u(1) = \beta, \quad x \in (0, 1), \quad 0 < \epsilon \leq \epsilon_0 \]

which has been presented in the studies of phase-locking in chains of weakly coupled oscillators as a continuum approximation (I). In [2], Reinhardt proposed the numerical treatment of linear singular perturbation difference problems using formal approximations and correctors. In this paper, we shall consider nonlinear difference equations. In this paper, by using singular perturbation technique, we construct two lower-order difference equations for the nonlinear difference equation (1) such that the sum of two corresponding solutions of these equations is the asymptotic approximation of that of (1). In this treatment, the main difficulty is to prove that the solution of a lower-order difference equation, which is called "outer solution", is uniformly close to the solution of (1) except the boundary layer. We have proved that by using the theory of exponential dichotomies this idea is motivated by the successful application of exponential dichotomies to multiple coupling in chains of oscillators [3].

In [1], N. Kopell and G. B. Ermentrout have proved a proposition by which we can determine where the boundary layer of the solution to the singularly perturbed problem is. In the following we suppose \( \epsilon > 0 \) and \( \beta(x) \) satisfy the boundary conditions, i.e., there is an interval \( J \) in which \( g'(u) > 0, \quad f'(u) < 0, \quad f'(u) \neq 0 \) and \( Q = f'(a) - f'(b) - \int_a^b \beta(x)dx > 0 \), for which the solution to (1) has a boundary layer on the L.H. side. For other cases, the solution has a boundary layer on the R.H. side, the similar procedure can be used to obtain the similar results.

2. REDUCED EQUATIONS AND OUTER SOLUTIONS

The reduced equations are

\[ \eta_k = \beta_k \eta_{k+1} - \omega_k \eta_k, \quad k = 1, \ldots, n, \quad \text{and} \quad \eta_{n+1} = \beta, \]

Firstly, we denote the difference \( u_k - v_k \) by a new variable \( \eta_k \), and derive a discrete equation of \( \eta_k \) and let \( \beta_{k+1} = \omega_{k+1} - \omega_k \). Then there is such a constant \( \Omega \) that

\[ \Omega = \omega_k + f(v_k), \quad k = 0, \ldots, n \quad \text{and} \quad v_{n+1} = \beta, \]

This solution exists provided that \( v_k \) stay in the region in which \( f(u) \) is monotone, and hence invertible.

Similarly, we can obtain the equations for such \( \Omega \)

\[ \Omega = \omega_k + f(u_k) + \frac{\epsilon}{h} \left( \begin{array}{c} g(u_{k+1}) - g(u_k) \end{array} \right) \]

\[ k = 1, \ldots, n \quad \text{and} \quad u_0 = \alpha, \quad u_{n+1} = \beta. \]

Using Taylor series, equations (3), (4) can be rewritten as

\[ 0 = f'(v_{k+1}) + \frac{\epsilon}{h} g'(v_{k+1}) (u_{k+1} - v_{k+1}) - g'(v_k)(u_k - v_k) + o(u_{k+1} - v_{k+1}) + o(v_{k+1} - v_k) + \frac{\epsilon}{h} o(\omega_k - \omega_{k+1}) \]

(5)

We note that the terms involving the form \( v_k - v_{k+1} \), are all \( O(h) \). Thus,

\[ \eta_{k+1} = \Lambda_k \eta_k \quad H(\eta_{k+1} \eta_k) + O(\epsilon) \]

(6)

where \( \Lambda_k = \frac{-\epsilon}{2g'(v_k)} \left( \begin{array}{c} 1 \end{array} \right) f'(v_{k+1}) + \frac{\epsilon}{h} g'(v_{k+1}) \left( \begin{array}{c} 1 \end{array} \right) \), \( H \) is at least quadratic in its variables and \( k = 1, \ldots, n \).

It is clear that if \( f'(u) \neq 0 \) for all \( u \in J \), then the absolute values of \( \Lambda_k \) are bounded uniformly away from 1. Now we wish to use this to show there are solutions \( u_k \) to (1) which stay arbitrarily close to any outer solution defined by (3) that satisfies \( v_k \in J \) in which \( f \) is monotone. For definiteness, we suppose \( f' > 0 \in J \).

Theorem 1: Let \( M_k \) be a sequence of \( n \times n \) invertible matrices, \( K \in Z \). Suppose that the linear difference equation \( Y_{k+1} = M_k Y_k \) has an exponential dichotomy on \( Z \) with constants \( K, \sigma \) and projections \( P_k \).

Suppose that, for each \( k \in Z \), \( H \) is at least quadratic in its variables and \( \eta_k \) is a bounded sequence. Then the implicit, nonlinear difference equation

\[ Y_{k+1} = M_k Y_k + H(Y_{k+1}, Y_k) + \epsilon \eta_k \]

(7)

has an unique solution \( Y_k \) such that for sufficiently small \( \epsilon \) and all \( k \in Z \),

\[ |Y_k| \leq 2K \left( 1 + \epsilon \sup_{k \geq 0} | \eta_k | \right) \]

(8)

Proof: The proof runs analogously to that in [4]

Theorem 1 can be used to difference equations defined for all \( k \in Z \) while our system (6) is defined only for \( 1 \leq k \leq n \). We must make a suitable extension for system (6) in order to use the exponential dichotomy theory. We note that if \( \omega_k \) are constants, then each outer solution \( v_k \) is also a constant, and hence the linearization \( A_k \) of (6) around \( v_{k+1} \) is also a constant. To consider (6) as an infinite system, we may define \( B_k = 0 \) for \( k \leq 0 \) and \( k \geq n+1 \). Then, if \( \omega_k \) are sufficiently close to a constant, then the linear homogeneous system associated with the suitable extension of (6) has an exponential dichotomy (4). Thus, we now establish the following theorem:
Theorem 2: Suppose that the above hypotheses. Then there is a solution to (1) that is arbitrarily close to the outer solution of (2) and

\[ u_k - v_k = O(\epsilon^k) \quad k = 1, \ldots, n \]

for sufficiently small \( \epsilon \).

3. CORRECTORS AND ASYMPTOTIC APPROXIMATION

Suppose that in our case a boundary layer behavior occurs at \( k = 0 \), and there is a number \( h_0 \) such that for \( 0 < h < h_0 \) we have the estimates:

\[ u_k - v_k < C \epsilon \]

where \( C \) is independent of \( h \) and \( \epsilon \). (\( h_0 \) can be determined by the roughness theorem for exponential dichotomies)

Analogously to [2], we set

\[ w_k = \epsilon^k \rho_k \quad k = 0, \ldots, n+1 \]

and we can obtain

\[ 0 = -F \rho_k + \frac{1}{h} G \rho_k, \quad k = 1, \ldots, n+1 \]

where \( \rho_0 \) is a parameter which will be determined by boundary conditions, \( F = f(0) \) and \( G = g(0) \)

By induction, the representation of \( \rho_k \) immediately can be obtained as follow

\[ \rho_k = \rho_0 \left( \frac{G}{F} \right)^k \quad k = 0, \ldots, n+1 \]

In order that \( \{v_k+w_k\} \) presents an asymptotic approximation we choose \( \rho_0 = \alpha - v_0 \). Finally we have

Theorem 3: Suppose that the above hypotheses. Then \( \{v_k+w_k\} \) is an asymptotic approximation of \( \{ u_k \} \) for \( k = 0, \ldots, n+1 \). The error satisfies the following estimates

\[ |u_k - (v_k + w_k)| \leq L \epsilon^{k-\frac{1}{2}} \]

where \( L \) is a constant independent of \( \epsilon \) and \( h \) and \( q > 1 \) is a constant.

Proof: Firstly, we have

at \( k = 0 \), \( u_0 - (v_0 + w_0) = (\alpha - v_0) - \rho_0 = 0 \).

at \( k = n+1 \), \( u_{n+1} - (v_{n+1} + w_{n+1}) \geq \beta(\alpha + e^{n+1} \rho_{n+1}) = e^{n+1} \rho_{n+1} \)

General, we have

\[ |\rho_k| \leq |\rho_0| \left( \frac{M}{h} \right)^k, \quad k = 0, \ldots, n+1. \]

Therefore, provided \( 0 < \epsilon < \epsilon_0 \), where \( \epsilon_0 = \frac{1}{h} \left( \frac{G}{F} \right)^{n+2} < 1 \), where \( q > 1 \), we can obtain the following estimates:

\[ |u_k - (v_k + w_k)| \leq |u_k - v_k| + |w_k| \]

\[ \leq C \epsilon + \epsilon^k |\rho_0| \left( \frac{M}{h} \right)^k \]

\[ \leq \epsilon^{k-\frac{1}{2}} (C + |\rho_0|), \quad k = 1, \ldots, n. \]

since \( \epsilon^k \left( \frac{M}{h} \right)^k \leq \epsilon \frac{M}{h} \leq \epsilon^{k-\frac{1}{2}} \) and we suppose \( |u_k - v_k| \leq C \epsilon \), where \( C \) is a constant.

This completes the proof.

4. EXAMPLES

The first example is

\[ eu'' + e^u' - \sin \frac{\pi u}{2} = 0, \quad u(0) = 0, u(1) = 0 \]

It's asymptotic approximation is

\[ u = \operatorname{Ln}(1 - \frac{2}{\pi} \cos \frac{\pi u}{2}) - \operatorname{Ln}(1 - \frac{2}{\pi} \exp(\frac{\pi}{2} - 1) \epsilon) + O(\epsilon) \]

and the solution has a boundary layer near the endpoint \( x = 0 \) (see Figure 1).

![Figure 1](image1.png)

Figure 1 (\( \epsilon = 0.01 \) and \( h = 0.02 \)) asymptotic solution, iteration solution and approximation solution

The second example we shall consider is

\[ eu'' + e^u' - (\frac{2}{\pi} \sin \frac{\pi u}{2}) e^u = 0, \quad u(0) = 0, u(1) = 0 \]

This nonlinear equation is due to O'Malley [5] and the variable fitting factor is used to find the numerical solution (6) (see Figure 2).

![Figure 2](image2.png)

Figure 2 (\( \epsilon = 0.01, h = 0.02 \)) asymptotic solution, iteration solution and approximation solution

REFERENCES


Abstract - We are interested in numerical methods for singular perturbation problems. Iterative algorithms for domain decomposition are considered. Numerical examples are presented for both one- and two-dimensional problems. We compare the performance of a serial iterative algorithm for domain decomposition and its parallel implementation.

1. INTRODUCTION

We are interested in numerical methods for singular perturbation problems. The solution of this problem exhibits a fine structure within small regions (boundary and interior layers) of the computational domain. The traditional numerical techniques for solving singularly perturbed problems require a fine mesh covering the whole domain in order to resolve these fine local details. These methods are inefficient, since the fine mesh is not needed in those parts of the domain where the solution has a moderate variation.

We present a numerical technique where the regions of rapid change of the solution are localized in space and therefore the refinement is applied locally (near boundary and interior layers). The construction of these special meshes is based on mesh generating functions (e.g. [1]). Except the grid refinement approach, our numerical method is based on domain decomposition. The domain decomposition technique provides a natural route to parallelism.

We introduce and analyze iterative algorithms for domain decomposition which reduce the given problem to sequences of boundary value problems on each subdomain. This numerical method is illustrated by solving singularly perturbed problems for elliptic equations. We consider the case of two subdomains. However, the results given also hold for more general situations.

Firstly the problem in the one-dimensional context is discussed. The same analysis is also generalized to the two-dimensional case. Numerical examples are presented for both one- and two-dimensional problems. Here we compare the performance of a serial iterative algorithm for domain decomposition and its parallel implementation.

2. ITERATIVE ALGORITHMS

We illustrate iterative algorithms for domain decomposition for the singularly perturbed one-dimensional elliptic problem

\[ L_\mu u(x) = \mu^2 \frac{d^2 u}{dx^2} = f(x,u), \quad x \in \Omega, \quad \Omega = (0,1), \]

\[ u(0) = u_0, \quad u(1) = u_1, \]

where \( \mu > 0 \) is a small parameter. The solution of (1a)-(1c) has boundary layers at \( x=0,1 \) and the size of boundary layers is of the order of \( h_\mu = \mu |ln(\mu)| / \mu \). For simplicity, we assume that the solution \( u(x) \) exhibits boundary layer only at \( x=0 \) (the "reduced" solution satisfies the boundary condition (1b) at \( x=1 \)).

We introduce the overlapping decomposition of the domain \( \Omega \) into two subdomains \( \Omega_1 \) and \( \Omega_2 \):

\[ \Omega_1 = (0, \delta), \quad \Omega_2 = (\delta, 1), \quad 0 < \delta < 1. \]

Consider two sequences of functions \((v^n), (w^n)\), \( n \geq 1 \), satisfying the problems:

\[ L_\mu v^n(x) = f(x,v^n), \quad x \in \Omega_1, \]

\[ v^n(0) = u_0, \quad v^n(\delta) = v_0, \]

\[ L_\mu w^n(x) = f(x,w^n), \quad x \in \Omega_2, \]

\[ w^n(\delta) = v^n, \quad w^n(1) = u_1. \]

We now construct two iterative algorithms. The first one, \( A_1 \), is the Schwarz alternating procedure. Here the boundary conditions \( v^n, w^n \) from (3) and (4), respectively, are defined by

\[ v^{n+1} = v^n(\delta), \quad w^{n+1} = w^n(\delta), \quad n \geq 1, \]

(the initial guess \( v^1 \) should be prescribed).

The second algorithm, \( A_2 \), is constructed using the interfacial problem

\[ L_\mu z^n(x) = f(x,z^n), \quad x \in \Omega_1, \quad \Omega_1 = (0, \delta), \]

\[ z^n(0) = v^n, \quad z^n(\delta) = w^n(\delta), \quad n \geq 1, \]

where \( 0 < \delta < \delta < 1 \). Here the boundary conditions from (3), (4) are determined by

\[ z^{n+1} = z^n(\delta), \quad v^{n+1} = z^n(\delta), \quad w^{n+1} = z^n(\delta), \quad n \geq 1, \]

(the initial guesses \( z^1 \) and \( v^1 \) are given).

Algorithm \( A_1 \) is a serial procedure, but algorithm \( A_2 \) can be carried out by parallel processing.
For algorithms $A_1$ and $A_2$ we have

**Proposition 1.** If $x > y$, then iterative algorithm $(3)$, $(4), (5)$ converges to the solution of problem $(1)$ with the linear rate $q$:

$$q = p(y)/p(x) < 1, \quad p(x) = \frac{\sinh(mx/\mu)}{\sinh(m(1-x)/\mu)}.$$

**Proposition 2.** Iterative algorithm $(3)$, $(4), (6)$ converges to the solution of problem $(1)$ with linear rate $q < 1$ provided $\Omega_1, \Omega_2, \Omega_{in}$ from $(2), (6a)$ fulfill

(a) $x - x_2^* \geq 2\mu/m, \quad x - x_1^* \geq 2\mu/m,$

or

(b) if $\mu$ is sufficiently small and $x - x_2^* = h_\mu, \quad x - x_1^* = h_\mu$ then $q = 0(\mu)$.

**Remark.** Iterative algorithm $A_2$ can be generalized straightforwardly to multiple-domain decompositions.

### 3. NUMERICAL EXAMPLES

We present the results of some numerical experiments using the iterative algorithms $A_1$ and $A_2$ described in previous section.

**Example 1.** We consider problem $(5)$, where $f(x,u) = 1 - e^{-u}$, $u_0 = 1$, $u_1 = 0$. Introduce a non-equidistant grid $w = \omega x^{\omega_y}$, where $\omega_x$ as in Example 1 and $\omega_y$ is a uniform one-dimensional mesh in $y$. The results are presented in Table 2. Here $N_x = 41$, $N_y = 25$ and $j = 21$, $k = 22$, $\Omega_{in} = (x_{k-1}, x_{k+1}) \times (0,1)$. If we implement algorithm $A_2$ on two parallel processors then $t_{A_2}/t_0 = 0.541$ $(j = 21, k = 22, K = 4)$.

<table>
<thead>
<tr>
<th>$\mu$</th>
<th>$h$</th>
<th>$K_{A1}$</th>
<th>$K_{A2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>0.05</td>
<td>0.1</td>
<td>0.01</td>
</tr>
<tr>
<td>0.05</td>
<td>59</td>
<td>35</td>
<td>9</td>
</tr>
<tr>
<td>0.01</td>
<td>7</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>0.001</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

Example 2. We consider the two-dimensional elliptic problem

$$\mu^2 \left[ \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right] = 1 - e^{-u}, \quad (x, y) = \Omega (0, 1) \times (0, 1),$$

$$u(0, y) = \Omega (0, y/2), \quad u(1, y) = 0, \quad y \in [0, 1],$$

$$u(x, 0) = 0, \quad \frac{\partial u}{\partial y}|_{y=0} = 0, \quad \frac{\partial u}{\partial x}|_{x=0} = 0, \quad y \in [0, 1].$$

Introduce a non-equidistant grid $w = \omega x^{\omega_y}$, where $\omega_x$ as in Example 1, and $\omega_y$ is a uniform one-dimensional mesh in $y$. The results are presented in Table 2. Here $N_x = 41, N_y = 25$ and $j = 21, k = 22$, $\Omega_{in} = (x_{j-1}, x_{j+1}) \times (0,1)$. If we implement algorithm $A_2$ on two parallel processors then $t_{A_2}/t_0 = 0.541$ $(j = 21, k = 22, K = 4)$.

### REFERENCES

SPECTRAL-FINITE ELEMENT SCHEME FOR NAVIER-STOKES EQUATIONS

GUO BEN-YU
Shanghai University of Science and Technology
Shanghai, 201800 P.R.C.

CAO WEI-MING
Shanghai University of Science and Technology
Shanghai, 201800 P.R.C.

Abstract. A spectral-finite element scheme is proposed for the lateral periodic and non-slip boundary problem of unsteady Navier-Stokes equations. The "inf-sup" condition is justified, and the convergence rates are presented.

I. SCHEME

Let \( Q \in \mathbb{R}^2 \) be a convex polygon and \( I = (0,2\pi) \). Let \( \mathcal{P} = \{(x,y) \mid x=x_1,x_2 \in [0,1], y \in I\} \). We consider Navier-Stokes equations as follows

\[
\begin{cases}
\frac{\partial u}{\partial t} + (u \cdot \nabla) u + \nabla p - \nu \Delta u = f, & \text{in } \Omega \times (0,T), \\
\nabla \cdot u = 0, & \text{in } \Omega \times (0,T), \\
u(x,y,t) = u_0(x,y), & \text{in } \Omega, \\
P(x,0,t) = \hat{P}(x,2\pi,t), & \text{for } x \in Q, \\
\end{cases}
\]

where \( u, P \) and \( \nu \) are the velocity, density, and viscosity respectively. \( f \) and \( u_0 \) are given functions with the period \( 2\pi \) for the variable \( y \). We consider the \( \ell \)-periodic and non-slip boundary conditions. It means that

\[
\begin{cases}
U(x,y,t) = 0, & \text{for } x \in \partial Q, y \in I, \\
U(x,0,t) = U(x,2\pi,t), & \text{for } x \in \Omega, \\
P(x,0,t) = P(x,2\pi,t), & \text{for } x \in \partial Q. \\
\end{cases}
\]

In addition, the pressure satisfies the normalization condition

\[
\| M(P) = \int_{\Omega} P(x,y,t) \, dx \, dy = 0. 
\]

Let \( C^0_p(Q) \) be the set of infinitely differentiable functions with the period \( 2\pi \) for the variable \( y \), \( H^m_p(Q) \) is the completion of \( C^0_p(Q) \) in \( H^m(Q) \) and \( H^0_p(Q) \) is the completion of \( C^0_p(Q) \) in \( L^2(Q) \) and \( H^0_0(Q) \) is the completion of \( C^0_p(Q) \) in \( H^0(Q) \). Let \( \mathcal{P}(Q) \) be the set of non-isotropic Sobolev spaces.

\[
\mathcal{P}(Q) = \left\{ \psi \in L^2(Q) \mid \| \psi \|_{H^m_p(Q)} < \infty \right\}. 
\]

The generalized solution of (1-3) is the pair \( (u(t),v) \in H_0^1_p(Q)^3 \) and \( p(t) \in L^2(Q) \) such that

\[
\begin{align*}
&J(u,v) + (u \cdot \nabla) u + \nabla p - \nu \Delta u = f, \\
&\nabla \cdot u = 0, \\
&u(0) = u_0, \\
&\| u \|_{L^2(Q)} = 1. 
\end{align*}
\]

We choose the trial subspace for the velocity \( \mathcal{V}_h \) of the form

\[
\mathcal{V}_h = \left\{ \psi \in C^0_p(Q) \mid \| \psi \|_{H^m_p(Q)} < \infty \right\}. 
\]

II. "INF-SUP" CONDITION AND CONVERGENCE

For convenience, we introduce firstly several non-isotropic Sobolev spaces. For \( r, s > 0 \),

\[
H^{r,s}_p(Q) = L^2(1,H^r_p(Q)) \cap H^s(Q) 
\]

and the norm

\[
\| \psi \|_{H^{r,s}_p(Q)} = \left( \| \psi \|_{L^2(1,H^r_p(Q))}^2 + \| \psi \|_{H^s(Q)}^2 \right)^{1/2}. 
\]

If \( r, s > 1 \), we define also

\[
H^{r,s}_0(Q) = \left\{ \psi \in H^{r,s}_p(Q) \mid \psi(0) = \psi(1) = 0 \right\}, 
\]

and the norm

\[
\| \psi \|_{H^{r,s}_0(Q)} = \left( \| \psi \|_{H^{r,s}_p(Q)}^2 + \| \psi \|_{H^{r,s}_0(Q)}^2 \right)^{1/2}. 
\]

We give two assumptions for \( \mathcal{V}_h \) and \( L_h \) as the
following:
(H1): \( \psi_h \) and \( \varphi_h \) satisfy the two-dimensional "inf-sup" condition, i.e., there exists a constant \( \beta > 0 \), independent of \( h \), such that (cf.(2))
\[
\sup_{\psi_h \in V_h, \varphi_h \in \mathcal{V}_h} \frac{\langle \varphi_h, \psi_h \rangle}{\| \psi_h \|_{L^2(\Omega)}} \geq \beta \| \varphi_h \|_{H^1(\Omega)},
\]
and \( \varphi_h \in \mathcal{V}_h \).

(H2): There exist \( \kappa > 1 \) and \( C > 0 \) such that for \( r > 1 \) and \( s > 1 \), is the generalized solution of (4), \( (U^*(t), P^*(t)) \) is its projection, i.e.,
\[
( \nabla (U(t)-U^*(t)), \varphi ) = 0, \quad \forall \varphi \in \mathcal{L}_h, \quad (6)
\]
then
\[
|U(t)-U^*(t)|^2 + |P(t)-P^*(t)|^2 \\
\leq C(h^{-1} + N^2) \left( \| U(t) \|_{H^{s-1}(\Omega)} + \| P(t) \|_{H^{r-1,s-1}(\Omega)} \right).
\]

Lemma 3 (Inverse inequality) There exists a constant \( C_0 > 0 \), depending only on the triangulation \( \{ C_h \} \), such that
\[
|v|^2 \leq (C_0 h^{-2} + N^2) \| \nabla v \|^2, \quad \forall \varphi \in \mathcal{V}_h.
\]

In order to get the convergence rates for the numerical solution of (5), we need only to estimate \( |U^*(t)-u(t)| \) and \( |P^*(t)-p(t)| \), where \( (U^*(t), P^*(t)) \) is defined in (6). By a similar analysis to that of [3], we can establish the convergence theorem as below.

Theorem Let \( (U, P) \) and \( (u, p) \) be solutions of

References

DETAILED NUMERICAL SIMULATION OF TWO-DIMENSIONAL IGNITION PROCESSES IN H$_2$O$_2$ MIXTURES

U. Maas, J. Warnatz

Institut für Technische Verbrennung der Universität Stuttgart, Pfaffenwaldring 12, 7000 Stuttgart 80, Federal Republic of Germany

Abstract

New numerical methods for the solution of stiff partial differential equation systems together with the availability of fast computers with high storage capacities now allow the global implicit simulation of instationary combustion processes in two space dimensions. Computations of ignition processes in hydrogen-oxygen mixtures are performed by solving the corresponding conservation equations (mass, energy, momentum, and species mass) using a detailed reaction mechanism (consisting of 37 elementary reactions) and a multispecies transport model. Thermal ignition is simulated by an additional source term in the energy conservation equation. Spatial discretization on a structured two-dimensional grid is adapted statically in two spatial directions leads to large differential-algebraic equation systems which are solved numerically by an implicit extrapolation method. Results are presented for the simulation of laser-induced thermal ignition of a hydrogen-oxygen mixture in a cylindrical reaction vessel. Due to the principal nature of the problem considered, application to many other problems seems to be possible, e.g. supersonic flow, chemical vapor deposition, atmosphere chemistry etc.

Mathematical Model

Mathematical simulation of chemically reacting multi-component compressible flow is performed by solving the corresponding system of conservation equations (Navier-Stokes equations) which may be written as [1]

\[
\frac{\partial \rho}{\partial t} + \text{div} (\rho \mathbf{u}) = 0 \tag{1}
\]

\[
\rho \frac{\partial \mathbf{u}}{\partial t} + \text{div} (\rho \mathbf{u} \otimes \mathbf{u}) + \text{grad} P = -\text{grad} \Phi + \mathbf{J} \tag{2}
\]

\[
\frac{\partial \rho}{\partial t} + \text{div} (\rho \mathbf{u}) = 0 \tag{3}
\]

\[
\frac{\partial \varepsilon}{\partial t} + \text{div} (\varepsilon \mathbf{u}) + \text{div} \left( \frac{\varepsilon + \rho}{2} \nu \text{grad} \rho \right) + \text{grad} \mathbf{J} = \frac{\partial \Phi}{\partial t} + \frac{\partial \mathbf{J}}{\partial t} \tag{4}
\]

where \( P \) = pressure, \( T \) = temperature, \( n_s \) = number of species, \( \omega_i \) = mass fraction of species \( i \), \( M_i \) = molar mass of species \( i \), \( \omega_f \) = molar scale rate of formation of species \( i \), \( h \) = specific enthalpy, \( p \) = density, \( \nu \) = velocity, \( \mathbf{J} \) = heat flux, \( \mathbf{J}_f \) = diffusion flux of species \( i \), \( \mathbf{J}_t \) = stretch tensor, \( \eta \) = source term for deposition of energy, \( t \) = time.

The equation system is simplified by restricting the problem to two-dimensional geometries (infinite rectangular column, finite cylinder). For cylindrical geometries (considered in the example below), there are two different boundaries, namely the axis of the cylinder and the vessel surfaces. Along the axis of the cylinder symmetry boundary conditions are used. The outer boundary conditions (i.e. those at the vessel surface) are simplified by assuming non-catalytic, adiabatic walls. Nevertheless, other boundary conditions can be introduced easily in order to account for interaction of surface processes with the gas-phase reaction, as was shown for one-dimensional geometries [2].

In order to allow a detailed description of the underlying chemical and physical processes, detailed transport as well as detailed reaction models are used. Transport coefficients are computed from molecular parameters, following the kinetic theory of gases and using the Curtiss-Fischel approach [3,4]. Thus, the transport coefficients depend non-linearly on temperature as well as mixture composition. Temperature-dependent properties, namely specific enthalpies and heat capacities, both depend on temperature and are computed from polynomial fits of data from the JANAF tables [5].

The chemical reaction mechanism used for the simulation of an induced ignition of a hydrogen-oxygen mixture consists of 37 elementary reactions [2], necessary for a detailed description of the complex processes which take place during the ignition process.

For the simulation of laser-induced thermal ignition of a hydrogen-oxygen mixture (shown below), a source term is introduced into the energy conservation equation. The energy density is assumed to decrease in axial direction and to have a Gaussian-like shape in radial direction (see [6] for details).

Transformation of the two-dimensional conservation equations into Lagrangian coordinates cannot be performed as easily as in the case of one-dimensional geometries, mainly due to the distortion of the grid point system [7]. Therefore in the approach of this work, the two-dimensional instationary conservation equations are solved in Eulerian formulation, taking into account spatial and temporal pressure and density fluctuations.

Solution Method

The partial differential equation system describing the reacting flow consists of \( n_s + 4 \) partial differential equations (continuity, momentum, energy, and species conservation equations). Together with the boundary conditions it forms an initial/boundary value problem which can be solved numerically. Several properties of this partial differential equation system require special solution methods. The main problems are orders of magnitude differences in the time and length scales, and in particular the stiffness introduced by the chemical kinetics.

Spatial discretization on a rectangular mesh using finite differences leads to a system of coupled ordinary differential and algebraic equations which can be solved numerically by a semi-implicit extrapolation method [8,9]. Due to the large differences of physical length scales (here particularly vessel diameter, flame front thickness, and diameter of the external energy source) adaptive gridding has to be used. In this work, the mesh is adapted statically in radial and axial direction, using a tensor product grid, i.e. a structured rectangular mesh. In the present computations, a 50 \( \times \) 40 mesh is used. Details of the adaptive gridding procedure can be found in [10].

Standard central difference approximations for the convective terms of the conservation equations can cause severe numerical instabilities ("over-shoots") in regions of high gradients and curvatures. Therefore, these terms have to be treated differently. Coupling of the discretization scheme to the flow direction by use of backward and forward differencing, depending on the direction of the flow ("upwind differencing") has the disadvantage that the accuracy of the difference approximation is only of the order of the grid point distances. This leads to a large amount of numerical diffusion -1 due to the flattening of steep gradients. Especially in the simulation of reacting flows, such steep gradients are present in the reaction zones (e.g. flame fronts), and numerical diffusion would falsify the results remarkably.

526
The approach used in this paper is based on the idea to use central difference approximations whenever possible and a modified upwind scheme, based on monotonicity preserving interpolation, otherwise. It is described in detail in [10].

Another problem arising in the numerical solution of the partial differential equation system is the resolution of shock fronts. In order to avoid severe numerical instabilities resulting if the shock is not resolved by the mesh, we apply an artificial viscosity term ("numerical diffusion") proposed by Richtmyer and Morton (see [7]) which spreads the shocks over a small number (typically 3) of grid points.

The system of ordinary differential/algebraic equations (resulting after spatial discretization) is solved using the semi-implicit extrapolation code LIMEX [8,9]. The Jacobian matrix required for the numerical solution has a block-non-diagonal structure. The dimension of the Jacobian is given by \( n_d \cdot n_a \cdot n_p \) where \( n_d \) is the number of grids in the radial, \( n_a \) the number of grids in the axial direction, and \( n_p \) the number of partial differential equations. For the example shown below, the dimension of the Jacobian is 20000. The computation of the Jacobian is performed numerically by difference approximation. In order to evaluate the Jacobian in a time saving way, use is made of the block-non-diagonal structure [10]. Due to the large dimension of the system, the solution of the linear equation systems (required by the time integration method) has to be performed by iterative methods (see [6,10] for details).

The simulation of the hydrogen-oxygen ignition (see below) takes about 50 hours on an IBM 3090; the code contains about 30,000 lines written in FORTRAN.

**Results**

As an example for the simulation of a chemically reacting flow, the model described above has been used to simulate a spatially two-dimensional ignition processes in a hydrogen-oxygen system with cylindrical geometry. In order to simulate induced ignition by a laser beam, thermal ignition is induced along the axis with a decreasing energy density (absorption of energy).

Spatial profiles of temperature and pressure in the reaction vessel at 1 ps, i.e. just after the external energy source has been turned off, can be seen in Figure 1. The temperature profile directly represents the spatial distribution of the ignition energy density. It decreases in axial as well as in radial direction. A similar behaviour shows the pressure, because the heating period is too short (1 µs) for the pressure to equilibrate all over the reaction volume. Thus, the pressure increase is approximately proportional to the temperature increase. The temporal development of the ignition process is shown in Figure 1, too. At the outer boundary of the ignition volume the pressure gradient causes the formation of a shock wave moving in the radial direction and a rarefaction wave moving towards the cylinder axis. After the shock wave has reached the vessel surface (after \( \approx 4 \) µs), it is reflected and forms a converging shock. Ignition (rapid temperature rise) occurs after a short induction period at locations where the amount of energy deposited during the heating period was high enough. Subsequently the flame front formed is moving in radial direction towards the outer boundary. Simultaneously, the flame propagates in the axial direction (in accordance with experimental results [11]). Two effects cause the flame propagation: Different induction times (depending on the local temperature) lead to a successive ignition along the axis, and at the same time a "regular" flame propagation takes place.
Fig. 1: Calculated time dependent profiles of pressure and temperature in an igniting stoichiometric hydrogen-oxygen mixture, cylindrical geometry, see [6] for details

Conclusions

It is possible to simulate ignition processes in hydrogen-oxygen mixtures using detailed chemistry and multi-species transport for two-dimensional geometries without simplifications like using a constant density approximation or the uniform pressure assumption. Operator splitting (which is a potential source of unreliable results) is avoided by using a fully implicit method. The method can be applied to even more complex reaction systems.

Because the hydrogen-oxygen system is an example for a chain branching ignition process, it allows an understanding of the complex interaction of chemical reaction and flow in systems of practical importance.

Furthermore, the methods described in this paper, allow the treatment of reactive flows other than those in combustion problems (e.g. supersonic flows, chemical vapor deposition etc.).

References

Abstract: In hypersonic flows of current and future technologies, interest complex reactive-diffusive processes occur in high-temperature multicomponent gas mixtures. The Couette flow lends itself as a convenient tool for testing the admittedly complicated physico-chemical models needed to understand and analyze such processes. In view of its simple geometry both the physical and mathematical aspects of the required models can be investigated through extensive and inexpensive parameter studies. Since the configuration has some features of both external flows (boundary layers on vehicles) and internal flows (in ducts and engines) such model studies yield information relevant to realistic prototype configurations. It is hoped thereby to contribute towards an understanding of the complex interaction of the various physico-chemical mechanisms involved and their relative importance so that some tractable forms can be derived. Extensive and inexpensive parameter studies. Since the configurational aspects involved in the flow fields in these configurations are highly complicated and depend on a large number of uncertain and incomplete set-ups, we may state that these configurations are highly complicated and depend on a large number of uncertain and incomplete set-ups and experimental studies. In view of the resurgence of interest in hypersonic for space transportation the basic zero-dimensional problems of high temperature gas dynamics are under intensive study in order to assist the R & D efforts for an efficient and economic design of the proposed prototype configurations. The challenging new aspects involved in the flow fields in these configurations are related to the multicomponent reactive-diffusive gaseous mixtures present in the vehicles and in the propulsion systems [1, 12]. Since the fundamental physico-chemical processes occurring in these configurations are highly complicated and depend on a large number of uncertain and incomplete set-ups of experimental data and theoretical models; it is felt worthwhile to investigate the effects of such inputs on some global quantities of practical interest such as heat transfer and skin friction at the vehicle walls. In hypersonic Couette flow, in particular, it is found that the catalytic efficiency of the atom and molecule as the reactive collisions occur (i.e., X + A or A + X) in Fig. 1. The mass-transport conditions in each case, will be considered.

II. PROBLEM FORMULATION AND SOLUTION

Fig. 1 illustrates the configuration and boundary conditions under consideration. We set up the general problem of a diatomic gas undergoing a dissociation/recombination reaction in a Couette flow without assuming constant Pr and Le, as was done by Clarke [3]. It consists of a system of coupled nonlinear ordinary differential equations with associated boundary conditions. The relevant boundary value problems (neglecting Soret and Dufour effects in dimensionless form are stated below. In, b, and other quantities. e.g., μ, k, etc.) evaluated at T∞ are used as reference values. Mathematically, the problem retains the essential nonlinear features arising from the physico-chemical processes of interest in hypersonics. We solve the boundary value problems by a multiple shooting method in the general case using thermophysical data from [5] - [8] and by Newton-Raphson iteration in the special case of chemical equilibrium [3].

Momentum:

\[ \mu \frac{\partial u}{\partial y} + \tau_w = \text{constant} \]  

Energy:

\[ \frac{\partial T}{\partial y} + \frac{D_{\text{rad}}}{T} \left\{ \frac{D_{13}}{(1 + \alpha_1)} \frac{d_1}{d_T} \right\} + \frac{E_{\text{ad}}}{\mu} \frac{d^2 \mu}{d_T^2} = N = \text{constant} \]  

Species:

\[ \frac{d}{d_T} \left\{ \frac{1}{(1 + \alpha_1)} \frac{d_1}{d_T} \right\} = \frac{\text{Dam}}{T^3} \left\{ \frac{2 \beta_1 \beta_2 + 1 + \alpha_1}{(1 + \alpha_1)} \right\} \]  

\[ \left\{ \frac{d_2}{d_T} - (1 - \alpha_2) \right\} \left( \frac{\mu_{\text{ad}}}{T^{\alpha_2 - 1}} \frac{\exp(-T_{\text{ad}} T)}{T} \right) \]  

We may note that \( \mu_{\text{ad}} \) and \( \mu_{\text{ad}} \) are in general different depending on the catalytic efficiency of the atom and molecule as the reactive collisions occur (i.e., X + A or A + X) in Fig. 1. The mass-transport conditions in each case, \( \alpha_1 = 0 \) and \( \alpha_1 = 1 \).
$G_e$ is related to the *dimensional* equilibrium constant

$$K_e = C_e T_e \exp(-T_p/T_e)$$

by the equation

$$K_e = (\rho G_e / \delta T_n) T_e \exp(-T_p/T_e).$$

The specific enthalpy of the diatomic molecule is taken to be [3], [4], [8], [9]

$$h_2 = \frac{G_e}{M_2} \left[ \frac{T}{2} + \frac{F_{vib}(T)}{T} \right].$$

where

$$F_{vib}(T) = \frac{T_p/T}{\exp(T_p/T_e) - 1}$$

is due to the harmonic oscillator vibrational energy of the molecule and contributes accordingly to molecular specific heat $c_{ps}$ and conductivity $k_e$ [3] - [5].

$F(T)$ in eqn. (2) represents the enthalpy difference

$$(h_2 - h_0) = D \cdot F(T) = \frac{T_p}{M_2} \left[ 1 + \frac{3T}{2T_p} \right]$$

$$\frac{(T_p/T_e)}{(\exp(T_p/T_e) - 1)}.$$

The IDG model [3], [4] is tantamount to assuming $F_{vib}(T) = 1/2$, $n_e = 0$, $D = 1$, and $G_e$ equal to some average value via a pressigned constant $\rho_p$ [4].

<table>
<thead>
<tr>
<th>Material</th>
<th>$T_e$ K</th>
<th>$T_p$ K</th>
<th>$\rho_p$ g/cm$^3$</th>
<th>$\rho_p(T_m)$ g/cm$^3$</th>
<th>$T_m$ K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nitrogen</td>
<td>3395</td>
<td>11261</td>
<td>130</td>
<td>137.4</td>
<td>2698</td>
</tr>
<tr>
<td>M$_1$ = 28, M$_2$ = 14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oxygen</td>
<td>2275</td>
<td>59355</td>
<td>150</td>
<td>170</td>
<td>1807</td>
</tr>
<tr>
<td>M$_1$ = 32, M$_2$ = 16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table I. Characteristic reference values used in the calculations. (Data based on values given in [6], [9].)

## III. RESULTS AND DISCUSSION

### A. General Model

The predominant influence for the cases considered by Clarke [3] (to be discussed later) is that due to the variation in Lewis number and the reaction rate coefficients. Variation of $P_r$ in the temperature range considered is not significant. On the other hand $Le$ can vary by more than a factor of two as a function of temperature and atomic concentration. In general we do not assume $P_r$ and $Le$ to be constant. We solve eqns. (1) - (3) using the mixture formulas [5] - [7] for transport coefficients and reaction rate coefficient data from Wray et al. cited by Vincente and Kruger [8], so that we have a standard dissociating diatomic gas in chemical nonequilibrium. Furthermore (assuming thermal equilibrium) we include the vibrational contribution both to the molecular enthalpy as well as specific heat. The ideal dissociating gas (IDG) of Lighthill [4] model assumes that half the vibrational states are excited and also that the equilibrium constant for the dissociation/recombination reaction is such that a characteristic constant density $\rho_p$ related to the atomic species concentration $n_{1a}$ at chemical equilibrium can be defined by

$$1 - n_{1a} = (\rho_p/\rho) \exp(T_p/T_e).$$

This $\rho_p$ is a function of temperature in general and has a relative maximum at a temperature $T_e < T$ [4], [8].

Profiles of velocity, temperature and atomic concentration and variation of the heat flux $H$ to the colder wall and the skin friction $S$ in the general case have been discussed in [10], [11] under b. c. 1 (with $U = 3.5$ km/s, $p = 1$ atm, $S = 1$ cm, $T_e = 1000$ K). The profiles of $T$ and $D$ indicate that temperature and concentration maxima can occur in the interior (and not necessarily at the hotter wall). This is known to occur in hypersonic boundary layers [12]. In fact it is even present in Couette flow at lower speeds [13] and is due to viscous dissipation. The temperature maxima move towards the hotter wall with increasing applied temperature gradient as molecular energy transport dominates viscous dissipation. The velocity profiles deviate slightly from linearity especially due to compressibility. A steeper rise in heat transfer rate is noted at temperatures ($\sim 2500$ K for oxygen and $\sim 4500$ K for nitrogen) where appreciable dissociation begins. This is due to the availability of more energy carriers, so to say. The skin friction $S$ on the other hand is not significantly affected by dissociation and increases mainly due to the viscosity increase with temperature (near linear velocity profiles contributing little to this increase).

### B. Vibrational Contribution

In Figs. 2, 3 we show the results of a comparative study of models for the vibrational contribution to the heat transfer and shear stress for nitrogen (Fig. 2 (a), (b)) and oxygen (Fig. 3). In both gases vibrational modes are activated around 800 K but oxygen is fully dissociated by 5000 K when nitrogen starts dissociating appreciably [12]. The relative differences DI, D$S$ (= absolute difference/ mean value) in heat flux $H$ and skin friction $S$ induced by the choice of models and approximations are illustrated in Figs. 2, 3. The reference values are those from the full equations of §II. The IDG model and three of its modifications are considered here. In all the modifications $\rho_p = 40/g/M_4$ (IDG value). In modifications two and three the equilibrium constant is that of IDG i.e.

$$K_e = \frac{d \rho_p}{M_2} \exp(-T_D/T_m),$$

each with a different (constant) $\rho_p$.

In the first modification

$$h_1 - h_2 = D (i.e., T_p > T_e)$$

and $K_e$ has the experimental (dimensional) value [8], viz.,

$$K_e = C_e T_e \exp(-T_p/T_e).$$

In the second modification

$$h_1 - h_2 = \frac{\delta T_D}{M_2} \left( 1 + \frac{T_e}{T_p} \right), \quad \rho_p = \rho_p(T_m)$$

i.e., vibrational contribution is averaged without neglecting $T/T_m$. In the third modification

$$\rho_p = \rho_p(T_m), \quad h_1 - h_2 = \frac{\delta T_p}{M_2} \left( 1 + \frac{T_m}{T_p} \right).$$

The interest in the last two modifications is that the mass assumption in IDG, viz., $F_{vib}(T) = 1/2$, happens to hold exactly at the temperature $T_e$ where $\rho_p$ has its relative maximum [14]. IDG uses a (smaller) representative value of $\rho_p$ [3], [4]. We see that the IDG model is quite good for oxygen (Fig. 3) and the relative difference in $H$ actually goes through zero at temperatures where the vibrational activity in oxygen is overtaken by viscous dissipation. Here IDG overestimates the heat transfer up to $\sim 2800$ K and understimates it at higher temperatures. But for nitrogen (which is vibrationally active even at higher temperatures) the IDG model is improved (cf. -DII gets smaller) in modifications two and three (Fig. 2 (a), (b)). IDG consistently understimates $H$ for nitrogen even for 7000 K. The deviations among the different modifications manifest themselves at temperatures where appreciable dissociation begins (Figs. 2 (a), (b)). Oxygen also shows this tendency for DII (Fig. 3).

The difference in $H$ both for nitrogen and oxygen is a few percent in magnitude but is found to be of opposite sign. Macroscopically the representation of vibrational energy is subsumed in $F_{vib}$ involving the ratio $T_p/T_e$ (cf. eqns. (6) - (8)). Hence the above distinction between the two gases is to be attributed to the large difference in their respective dissociation temperatures (cf. Table 1).
20. Under b. c. d. c. 3 the dimensional values of the global quantities \( H, S, \) and temperature \( T(1) \) at the upper insulated, noncatalytic wall and \( L = \alpha(1)/\alpha(0) \) were computed [11] as functions of the lower wall temperature \( T_u \) up to 1000 K. Since the lower wall quantities are used in our non-dimensionalization the computed results in dimensionless form are not easy to compare in this case. Hence the actual dimensional values were used. Heat transfer, skin friction and \( T \), are found to increase with \( T_u \) almost linearly while the concentration ratio varies very steeply in view of the negligible atomic species concentration at the low \( T_u \) values [11].

D. Transport coefficients

We now turn to the special case [2] of chemical equilibrium under b. c. 1 to illustrate the effect of transport coefficient variation. We regard here \( Pr \) and \( Le \) as dimensionless representatives of diffusive transport. It may also be noted that the equilibrium case serves as a useful upper bound for estimating heat transfer and was also used for nitrogen.

![Comparison of the vibrational contribution in IDG and its modifications vs. the general case under b. c. 1: Relative difference in (a) heat flux \( H \) and (b) skin friction \( S \) for nitrogen.](image1)

![Comparison of the vibrational contribution in IDG and its modifications vs. the general case under b. c. 1: Relative difference in heat flux \( H \) for for oxygen.](image2)

Relative differences in \( S \) are quantitatively even smaller than in \( H \) since skin friction is not significantly affected by dissociation or vibration but even here we find qualitative differences between the two gases. For oxygen the relative difference in \( S \) is found to steadily decrease with temperature whereas for nitrogen (Fig. 2 (b)) the different modifications show mutual deviations at temperatures where dissociation begins but vibrational modes are still active. Modification no. 1 slightly improves on IDG here.

C. Boundary conditions

Increase in heat flux \( H \) under b. c. 2 with \( \theta \) is similar to that under b. c. 1 correlating with the rapid increase of dissociation for nitrogen at about 4500 K. Skin friction increases rather slowly as under b. c. 1 [11].

![Chemical equilibrium \( O_2 - O \) flow under b. c. 1. Variation of (a) heat flux \( H \) and skin friction \( S \) and (b) temperature overshoot \( T_{max} \) and corresponding velocity \( U_{max} \) with \( M \) and \( Le \).](image3)
as such in Space Shuttle comparisons [12]. Setting Pr = 3, a,  = 4/3, p = 1 atm and assuming = f( ) as in [2] we compute the velocity-temperature profiles for given = at equilibrium. We observe that = ( ) in the flow field is compatible with the catalytic wall in b. c. 1. Clarke showed that setting constant values for Pr and Le considerably reduces the nonlinearity in the problem and allows integration of the (modified) eqn. (2). In the equilibrium case he used the IDG model for = whereas we test other models as well. Fig. 4 (a) shows that the heat flux in contrast to the skin friction is quite sensitive to variations of Le, especially at lower Mach numbers. Skin friction shows a slight dependence on Le at high Mach numbers. Although both II and S increase with Ma as to be expected their variation with Le is different. The latter difference with respect to Le may be interpreted as due to the role of diffusion (proportional to Le). Diffusion of species increases energy transport and also tends to equalize momentum, i.e., to reduce velocity gradients in the fluid layers. Thus with increasing Le skin friction is reduced. But this effect is only slight and confined to the higher Mach numbers.

As indicated in the inset of Fig. 4 (b) temperature overshoots, i.e., temperatures higher than at the hotter wall, can occur in the flow field. In Fig. 4 (b) we show the maximum temperature Tmn and the corresponding Umn as functions of Ma and Le. Variation of Umn shows (since velocity profiles are almost linear) that Tmn moves towards the middle of the flow field at all Le. Thus temperatures at which ionization and radiation would become important factors for electromagnets and heat transfer e. g. Tmn ~ 10000 K at Ma = 18) can arise in the flow field. Under such circumstances the base problems have to be reformulated appropriately. It is found that the temperature overshoots start to occur in nitrogen at lower Ma than in oxygen but are then overtaken by oxygen at higher Ma. This has corresponding implications for air in which nitrogen is the more abundant component.

II. Equilibrium constant

As another example of the use of Couette flow as a test case for studying physico-chemical models we show in Fig. 5 the relative differences for oxygen in II (Fig. 5(a)) and in Le (Fig. 5(b)) as functions of Le and Ma with reference to the equilibrium constant in terms of ey used in the IDG model [3] and that of Wray et al. cited by Vincenti and Kruger [8]. As already mentioned, the equilibrium concentration in the general case is given in terms of ey by

\[
\alpha_1(\text{equilibrium}) = \alpha_1(\text{eq}) = \sqrt{\frac{G_1\phi_{10}^{(2)}}{4 + G_1\phi_{10}^{(2)}}} \exp(-T_0/71) \tag{10}
\]

The differences are a few percent here and are found to be even smaller for S and Umn. Thus the IDG model is in this sense quite useful especially for oxygen.

IV. CONCLUSIONS

IDG model offers a numerically acceptable approximation for nitrogen and oxygen for heat transfer and skin friction but the intrinsic differences between the two gases are nevertheless qualitatively discernible. Although the computations tested specific assumptions and approximations in particular examples the results shown in the simple cases considered illustrate the feasibility of assessing various physico-chemical models and role of parameters with regard to their influence on practically relevant quantities of interest in hypersonics. Such efforts should prove especially valuable in optimal use of experimental and numerical efforts through identification of trends to be expected and selection criteria for useful models. In view of the relative simplicity of the configuration geometry efficient numerical means can be devised to analyze even elaborate and complicated physico-chemical models.

REFERENCES


Fig. 5 Chemical equilibrium O2 - O flow under b. c. 1: Relative differences in (a) heat flux II and (b) Tmn between the IDG and Wray value for the equilibrium constant.
INTERFACE MORPHOLOGIES DURING LASER MELTING OF THIN SILICON FILMS

S. R. CORIELL and G. B. McPADDEN
National Institute of Standards and Technology
Gaithersburg, MD 20899 U.S.A.

ABSTRACT — Thin silicon films on a cooled substrate are often found to develop highly nonlinear interface morphologies upon radiative heating. We develop a boundary integral representation of the thermal field, and obtain numerical solutions for nonplanar solid-liquid interfaces.

I. INTRODUCTION

Radiative heating of silicon is an important processing technique for silicon wafers, especially for making silicon on insulators which are dielectrically isolated from a substrate, and also for annealing of buried layers created by ion implantation. Two-phase mixtures of solid and liquid silicon have been observed to form during the laser processing of thin silicon films on substrates such as fused silica. Furthermore, it has been observed that the morphology of the interfaces separating liquid and solid phases may either be planar or corrugated, depending upon the values of the experimental parameters. The experimental results for the two-phase silicon mixture appear to be fully consistent with the fact that liquid silicon has a higher reflectivity than solid silicon. Upon heating the silicon-layer above its melting point, supercooled liquid can form adjacent to superheated solid silicon. This thermal configuration gives rise to a two-phase mixture, and leads to conditions under which planar solid-liquid interfaces within the mixture become morphologically unstable.

In this paper we extend the linear analysis by Kurtze and Jackson [Journal of Crystal Growth 71 (1985) 385] for a periodic array of silicon lamellae, and develop a numerical technique to treat the fully nonlinear free boundary problem.

II. NUMERICAL METHODS AND RESULTS

Jackson and Kurtze consider a two-dimensional model for heat flow in the film consisting of two dimensionless diffusion equations with source terms that result from the imposed heat fluxes $J$ through the plane surfaces of the film; in each phase the source term is taken to be a constant. The diffusion equations thus assume the form

$$\frac{\partial \theta_L}{\partial t} = \frac{\partial^2 \theta_L}{\partial z^2} + \frac{\partial^2 \theta_L}{\partial y^2} - J_L,$$

for the liquid phase, and, for the solid phase,

$$\frac{\partial \theta_S}{\partial t} = \frac{\partial^2 \theta_S}{\partial z^2} + \frac{\partial^2 \theta_S}{\partial y^2} + J_S; \quad (2)$$

here the quantities $\alpha_L$ and $\alpha_S$ are the thermal diffusivities of the liquid and solid, respectively.

At a crystal-melt interface the Gibbs-Thomson condition,

$$T_L = T_S = T_M - \gamma L K,$$

(3)

gives the equilibrium melting temperature at a curved interface, where $T = \gamma L K$ is a capillary coefficient, $\gamma$ is the surface tension, $L$ is the latent heat of fusion per unit volume of solid phase, and $K$ is the mean curvature of the interface. Conservation of heat at the interface requires

$$k_L \nabla T_L \cdot \hat{n} - k_S \nabla T_S \cdot \hat{n} = -L \nu_n,$$

(4)

where $\hat{n}$ is the unit normal vector at the interface pointing into the liquid phase. $k_L$ and $k_S$ are thermal conductivities, and $\nu_n$ is the normal velocity of the interface.

Steady solutions to the nonlinear governing equations may be expressed in terms of boundary integrals, which also provide an accurate and efficient computational procedure. This effectively reduces the dimensionality of the problem from two to one, and allows the calculation of interface shapes that are not easily expressed as a single-valued function. Implementation of the boundary integral technique for the treatment of arbitrarily shaped interfaces is facilitated by using a relative arclength representation for the solid-liquid interface. We set $e = s/S_T$, where $s$ is the arclength along the interface and $S_T$ is the total arclength of the interface over a full period. We then describe the interface parametrically as the set of points $(x(e), y(e))$ for $0 < e < 1$; the functions $y(e)$ and $x(e) - e$ are both periodic functions of $e$. Given an interface shape, the temperature at a point $z' = z + y'$ in the liquid can be written in the form

$$T_L(z') = \frac{1}{2} J_L (y - 7_T) + \int_s \sigma_L \frac{\partial G_L(x,z')}{\partial n} \, ds,$$

(5)

where

$$G_L(x,z') = \frac{1}{2\pi} \log |\sin\pi(x - z')| + \frac{1}{2\pi} \log |\sin\pi(z - z'^*)| \quad (6)$$

is a periodic Green's function with reflection symmetry about the line $y = 7_T$; here the image point $x'^*$ is given by $x'^* = x'^* - i[(2\pi - L)y']$. The normal derivative and the integration in Eqn. (5) are performed with respect to the unprimed variables, and $s$ is the arclength along the interface curve $\gamma$ over a single period. The outward normal to the curve $\gamma$ is given by the vector $(y_0 - y)/(2\pi)$. The unknown dipole distribution $\sigma_L$ is determined by the Dirichlet boundary conditions (3) for $T_L$. An integral equation of the second kind for $\sigma_L$ is obtained by letting $x'$ tend to a point on the boundary $\gamma$; this equation is discretized and inverted to give an approximation to the dipole distribution $\sigma_L$. Having found $\sigma_L$, we compute the normal derivative $\partial T_L/\partial n$ required in the conservation of heat condition at the interface, Eqn. (4). A similar procedure is used to compute $\partial T_S/\partial n$. In general, for arbitrary interface shapes, the heat flux equation is not satisfied, so iteration of Eqn. (4) using Newton's method is applied to find the unknown interface $(x(e), y(e))$. For each updated guess of the interface shape, the entire integral equation solution method is repeated in order to find new values of $\partial T_L/\partial n$ and $\partial T_S/\partial n$. Provided the initial guess is sufficiently close, the procedure quickly converges to an interface shape consistent with all equations and boundary conditions.

We choose to parameterize the interface by the tangent angle $\phi(e)$,

$$\phi(e) = \tan^{-1} \left( \frac{dy}{dx} \right),$$

(7)

and express $\phi(e)$ as a Fourier series,

$$\phi(e) = \sum_{n=0}^{M} a_n \cos 2\pi n e + b_n \sin 2\pi n e.$$

(8)

Given the function $\phi(e)$, the interface shape $(x(e), y(e))$ can be computed from the $\phi(e)$ and the initial values $x(0) = 0$ and $y(0)$ by quadrature.

For numerical purposes we retain a finite set of Fourier coefficients for $\phi(e)$ by truncating the infinite series at $n = M$. The boundary integral equations are then discretized using the trapezoidal rule; this results in a numerical approximation with spectral accuracy. The
heat conservation boundary condition (4) is also discretized to give a system of nonlinear equations in the unknowns $a_n$, $b_n$, and $g(0)$.

We have used values appropriate to silicon in our numerical calculations. Periodic solutions having wavelength $\lambda$ are computed most efficiently by choosing the computational domain to have length $\lambda/2$ in the $z$-direction, with no-flux boundary conditions applied as $z = 0$ and $z = \lambda/2$, we achieve this in practice by imposing this symmetry on the Fourier coefficients, setting $a_n = 0$. On this domain one may compute solutions which are periodic with wavelength $\lambda$ and also solutions which are higher harmonics with wavenumbers which are integer multiples of $\omega = 2\pi/\lambda$.

![Sequence of interface shapes tracked along the $\lambda$ solution branch.](image)

Figure 1. A sequence of interface shapes calculated using a continuation method starting with a planar shape at the onset of instability of a perturbation with wavelength equal to $\lambda$. We define the amplitude $A$ of an interface shape to be the square root of the sum of the squares of the Fourier coefficients for the tangential angle, so that the amplitude is zero for the planar state. As the solution branch is traced out to larger amplitudes, nonlinear effects cause higher order harmonics (in particular the first harmonic) to develop. The fundamental component gives way to the appearance of its first harmonic, and ultimately disappears. The computational domain also allows solutions with wavenumbers that are integer multiples $n\omega$ of the fundamental wavenumber $\omega$; these solutions bifurcate from the base state at progressively higher values of the power $J_L$. The result of tracking several such solution branches is shown in Fig. 2, which is a plot of the amplitude of the steady state interfacial shapes as a function of the power $J_L$. We also calculate numerically the linear stability of each of the computed nonlinear steady states. On the plot, stable nonlinear interfacial shapes are represented by solid curves while the dashed curves represent unstable interface shapes. The planar state first loses stability to the fundamental mode ($n = 1$) at $J_L = 1.7 \times 10^5 \, K/cm^2$, which bifurcates supercritically. This fundamental solution branch increases in amplitude with increasing power until it reaches a limit point near $J_L = 3.4 \times 10^5 \, K/cm^2$, after which it increases in amplitude with decreasing power; this branch loses stability at the limit point. It terminates at finite amplitude at a secondary bifurcation point on the nonlinear solution branch corresponding to solutions with period $\lambda/2$ ($n = 2$). This $\lambda/2$ family of solutions bifurcates subcritically from the planar solution at $J_L = 3.3 \times 10^5 \, K/cm^2$, and the family is initially unstable to two types of perturbations; for small amplitudes, the unstable disturbances are approximately sinusoidal with wavenumbers $\omega$ and $2\omega$. The $\lambda/2$ family reaches a limit point near $J_L = 2.3 \times 10^5 \, K/cm^2$, where it sheds one mode of instability, and then encounters the secondary bifurcation point with the fundamental solution branch, where the remaining mode of instability is lost, the $\lambda/2$ family is then stable for increasing values of the power. The $\lambda/3$ family bifurcates subcritically at $J_L = 7.0 \times 10^5 \, K/cm^2$, and initially has three modes of instability. The $\lambda/3$ family also has a limit point at $J_L = 3.5 \times 10^5 \, K/cm^2$, which is followed by two secondary bifurcation points, after which it continues to larger power as a stable solution branch. To avoid further complicating the figure, only the primary solution branches are shown, and the solution branches issuing from these secondary bifurcation points are omitted.

![Bifurcation diagram of interface amplitude $A$ versus the power $J_L$ for solutions branches corresponding to integer multiples $n\omega$ of the fundamental wavenumber $\omega$.](image)

Further primary solution branches for the families with periods $\lambda/4$ through $\lambda/8$ are also shown in this figure; their points of bifurcation occur off-scale. Although the separation between these bifurcation points from the planar state are increasing well-separated, they are also progressively more subcritical, and have limit points that all occur within the range shown in the figure with amplitudes at the limit point in the range $1.5 < A < 2$. In each case the family with period $\lambda/n$ initially has $n$ modes of instability, and the trend of shedding instabilities at secondary bifurcation points once the limit point is passed also seems to hold; to avoid further complicating the figure the secondary bifurcation points are not shown for the primary modes with $n > 3$. The primary mode with $n = 4$, for example, regains stability at $J_L = 6.6 \times 10^5 \, K/cm^2$, as indicated by a dot in the figure.

We are indebted to K. A. Jackson for suggesting this problem. This work was conducted with the support of the Microgravity Science and Applications Division of NASA, and the Applied and Computational Mathematics Program of DARPA.
Abstract. Wave-breaking events on the surface of the ocean entrain air, which creates a layer of bubbles immediately beneath the surface. The sound speed in the layer increases monotonically with depth, forming a surface duct in which sound may propagate in the form of normal modes. Since sound is currently being used to study a number of surface processes, including gas transfer across the air-sea interface, the effect of the profile on the observations is of some concern. An analytical theory of sound propagation in surface ducts has recently been developed, based on an inverse-square profile, which provides an exact solution for the acoustic field in terms of a sum of normal modes. On the basis of the theory, it is possible to obtain the solution of certain inverse problems; for example, once the parameters of the profile have been prescribed, the source depth can be estimated from the width of the modulation structure that appears in the acoustic spectrum at the receiver. The theory also provides an interpretation of the acoustic signatures of wave-breaking events observed in the La Perouse and FASINEX experiments.

I. INTRODUCTION

When an ocean wave breaks, air is entrained and forced below the sea surface, where it fragments into a large number of micro-bubbles. Acoustically, the bubbles have two important effects. For a few milliseconds after the instant of closure a bubble rings, that is to say, it oscillates in the radial or breathing mode, hence acting as a very effective acoustic source, this being the mechanism which is responsible for much of the sound produced by breaking waves, and the presence of the large population of bubbles reduces the speed of sound immediately below the surface by as much as 20 m/s. The sound speed profile down through the bubble layer is monotonic increasing, which gives rise to upward refraction, thus creating the condition necessary for the formation of a surface waveguide. The sounds from breaking waves are trapped in this bubbly waveguide and hence undergo dispersion, which manifests itself in the form of well-defined features in the spectrum observed at a receiver in the duct

An analytical model of sound propagation in the ocean-surface bubble layer has been developed [2], based on the so-called inverse-square profile:

$$\frac{1}{c^2(z)} = \frac{1}{c^2} \left(1 + \frac{d^2}{z^2}\right), \quad z \geq z_c,$$

where $c(z)$ is the speed of sound as a function of the depth coordinate, $z_c$ the parameter $d$ provides a measure of the effective depth of the duct, and $c$ is the asymptotic value of the sound speed in the limit of infinite depth. In the coordinate scheme of Eq. (1), the origin of $z$ lies above the sea surface, which falls at $z = z_o$. Below the surface, the ocean is treated as a semi-infinite medium, and the surface itself is assumed to be a plane, pressure-release boundary. The most significant features of the new theory are described briefly below, with emphasis placed on their relevance to the interpretation of wave-breaking spectra.

II. PROPAGATION IN AN INVERSE-SQUARE DUCT

The Helmholtz equation can be solved exactly for the harmonic field from a point source in an inverse-square channel bounded above by a pressure-release surface and extending below to infinity. The solution takes the form of a branch line integral, representing short-range radiation, and an infinite sum of normal modes. Only the modal component of the field is significant in the present context. Immediately below the sea surface each mode shows oscillatory behavior, down as far as the extinction depth. The oscillatory region is where the modal energy is concentrated. Below the extinction depth the oscillations cease and the mode exhibits an exponential decay to zero. The oscillatory and evanescent regions are illustrated in Fig. 1.

Fig. 2 shows the magnitude of several modes at a fixed depth as a function of frequency. In this example, the parameters of the inverse-square profile are representative of the surface bubble layer in the FASINEX experiment [1]. Each mode occupies a finite bandwidth, within which it displays several well-defined peaks. It is evident that on performing a coherent modal synthesis, to obtain the power spectrum of the field, interference between the overlapping regions of these modes will occur. Thus, the resultant spectrum will inevitably have a complicated structure of peaks and troughs. It is reasonable to suppose that a measured spectrum of sound (from breaking waves or any other source) in a surface duct should be similarly complex.

In the analysis of such spectra, the mode drop-out frequency would seem to be a useful interpretive tool, and indeed it is, as Farmer and Vagle [1] have demonstrated; but, at best, it is rather crude. This observation is illustrated in Fig. 2, where it can be seen that the drop-out frequency, depicted by the vertical lines, falls close to the principal
maximum in the odd-order modes but is almost coincident with the highest zero in the even-order modes. In general, the drop-out frequency is not an accurate indicator of the position of the principal maximum in the spectrum of a mode, differing perhaps by 10% or so, as in modes 8 and 12 in Fig. 2. When a number of modes are superposed, this discrepancy is compounded by the fact that each mode shows secondary maxima which may be comparable in magnitude with the principal maximum, making interpretation of the peaks in the spectrum doubly difficult. Both problems are eliminated by computing the full theoretical spectrum for comparison with the experimental observations.

III. WAVE-BREAKING SPECTRA

Farmer and Vagle [1] recorded the sound of breaking waves at two locations, identified as La Perouse and FASINEX, with a shallow hydrophone at a depth of 14 m and 24 m, respectively. A surface bubble layer was present in both cases, its depth, as measured by the parameter $d$ in Eq. (1), being a factor of 2.5 greater in FASINEX than La Perouse. The observations were made over the frequency band from 1 to 20 kHz.

The observed spectra from the two locations are different in structure. That from La Perouse shows half a dozen well-defined, relatively narrow peaks, whereas the FASINEX spectra show several broad bands of energy separated by distinguishable gaps. These features are believed to be genuine, rather than artifacts of the instrumentation, because they are repeatable over a number of wave-breaking events but with minor variations due to the slowly changing nature of the bubbly surface duct (D. M. Farmer, private communication, September 1990).

![Figure 3a](image1)

![Figure 3b](image2)

**Figure 3.** Inverse-square spectra (solid lines) for a) La Perouse and b) FASINEX. The cross-hatching indicates observed spectral peaks in a) and spectral bands in b). The vertical dotted line in a) depict the drop-out frequencies of the indicated modes.

Fig. 3a shows the spectrum (solid line) of a wave-breaking event at La Perouse, as computed from the inverse-square theory using channel parameters obtained from measurements [1] of the sound speed profile through the bubble layer. For comparison, the cross-hatched, horizontal stripes indicate the spectral maxima observed by Farmer and Vagle [1]. Notice the isolated spectral maxima and the fidelity of the match between theory and experiment.

Similarly, Fig. 3b shows the inverse square theory (solid line) compared with experimental observations (cross hatching) for FASINEX, the deeper of the two-surface channels. Again, the theoretical spectrum contains well-defined maxima, but in this case clustered into groups, or bands, separated by regions of little or no energy. The positions of the bands closely match the experimental observations [1], as indicated by the cross hatching.

The detailed agreement between the inverse-square theory and experiment suggests strongly that in La Perouse and FASINEX the observed spectral structure is an effect of propagation through the upward-refracting surface bubble layer. An alternative proposal [3], that the spectral features are an effect of the source (non-linear bubble oscillations), may be valid in other situations, where the surface bubble layer is not well developed, but in the La Perouse and FASINEX data set the propagation does appear to dominate the spectrum.

IV. SPECTRAL PEAKS AND SPECTRAL BANDS

The two spectra in Fig. 3 show well-defined maxima that are obviously related to the spectral peaks in the individual modes. In Fig. 3b, these peaks are strongly modulated, to form bands which are separated by gaps, where little energy exists. In fact, a similar modulation is present in Fig. 3a, but so much slower that it is barely noticeable. (The spectral period of the modulation is inversely proportional to $d$, the effective depth of the channel, which is larger in FASINEX than in La Perouse by a factor of 2.5).

Apart from $d$, the modulation also depends on the source depth, $z'$. If $Afo'$ is the spectral period of the modulation (i.e. the distance between the bands), then the following approximate relationship holds:

$$\Delta f' = \frac{C_0}{2d \ln \left( \frac{\rho}{\rho'} \right)}.$$  \hspace{1cm} (2)

It follows that, once the parameters of the profile are known, it is possible to perform an inverse calculation to determine the depth, $z'$, of the source from a measurement of $Afo'$.

On applying this argument to the FASINEX data, a source depth of 1.5 m is obtained, and this is the value that was used in the computation of Figs. 3a and 3b. Now, this is unexpectedly deep for acoustically active bubbles. The bubble layer is known to extend down to several metres, but most of the constituent bubbles are mature and hence quiescent. It is, perhaps, possible that the roughness of the sea surface and the non-uniformity of the bubble layer could be factors in leading to an exaggerated estimate of the depth of the bubble sources. Although this cannot be discounted, the fact that such good agreement between theory and experiment is observed in Fig. 3 suggests that the bubbly duct does act as a deterministic waveguide with a planar-pressure-release boundary, at least over the short ranges considered here. Such behavior would not be expected if non-uniformity in the channel, for whatever reason, were a significant factor.

V. CONCLUDING REMARKS

The inverse-square theory of sound propagation through an upward-refracting surface channel shows excellent agreement with observations [1] of wave-breaking spectra obtained at two locations with a near-surface hydrophone. This agreement supports the conclusion that the spectral structure present in the observed wave-breaking signatures is introduced as the signal propagates through the bubbly surface duct.

REFERENCES

MARCHING TECHNIQUES BASED ON ELLIPTIC WAVE EQUATIONS

George H. Knightly* and Donald F. St.Mary†
Center for Applied Mathematics and Mathematical Computation
Department of Mathematics
University of Massachusetts
Amherst, MA 01003, USA

Abstract: A class of methods for marching acoustic waves forward in range is presented, based on a far field elliptic approximation to the Helmholtz equation. The stability of the methods is analyzed and two particular schemes are examined in more detail. Explicit stability criteria are obtained for these two schemes, amounting to easily verified restrictions on the step sizes. The stability criteria are affirmed in sample computations.

I. INTRODUCTION

We are concerned here with the problem of determining the acoustic wave due to a source of given frequency, \( \omega \), in the ocean waveguide. We suppose the solution is known (e.g., by measurement) out to some range, \( r_0 \), and we wish to propagate the solution forward to larger ranges. To simplify the discussion we consider here only the azimuth-independent case and utilize the range coordinate \( r \) and depth coordinate \( z \). Thus, we investigate the following problem for the Helmholtz equation with variable wave speed \( c(r, z) \), pressure release boundary at the surface \( z = 0 \) and hard bottom \( z = B \).

\[
\nabla^2 p + \kappa^2(r, z)p = 0, \quad r_0 < r, \quad 0 < z < B,
\]

\[
p(r, 0) = 0, \quad \frac{\partial p}{\partial z}(r, B) = 0, \quad r_0 < r \quad \text{(1b)}
\]

\[
p(r, z) \text{ given,} \quad r < r_0 \quad \text{(1c)}
\]

where \( \kappa(r, z) = \omega / c(r, z) \).

When backscatter is unimportant, parabolic equation methods are widely used (e.g., see [5] and references cited therein) to march the solution forward in range. Here we continue some investigations [1, 3, 4, 6] into marching with a far field elliptic approximation that retains the potential to include backscatter.

II. MARCHING SCHEMES

In this section we develop a family of marching schemes for problem (1) (i) introducing Padé rational approximations for an operator in (3a), (ii) discretizing the resultant approximate problem and (iii) specifying a technique for solving the discrete problem.

We first rewrite the far field elliptic equation (3a) as

\[
\left( \frac{\partial}{\partial r} + ik_0 \right)^2 w - (ik_0)^2 \phi + 1 + \frac{\partial^2}{\partial z^2} w = 0 \quad \text{for the operator}
\]

\[
1 + h = (\sqrt{1 + h^2}),
\]

where

\[
h = \phi + \frac{1}{k_0^2} \frac{\partial^2}{\partial z^2}.
\]

We apply the chosen approximation in (5) and operate with \( Q^2 \) to obtain

\[
Q^2 w_{rr} + 2ik_0 Q^3 w_r + k_0^2 (P^2 - Q^2) w = 0.
\]

We specify mesh points by choosing a range step size \( dr \) and a depth step size \( dz = B/M \), for a chosen integer \( M \). Set

\[
w_m = w(r_0 + mdr, 0), \quad m = 0, 1, \ldots, M + 1; \quad n = -1, 0, 1, 2, \ldots
\]

and

\[
w_n = (w_n^1, \ldots, w_n^M)^T, \quad n = -1, 0, 1, 2, \ldots
\]

The boundary conditions are discretized as \( w_n^0 = 0 \) and \( w_n^{M+1} = w_0 \).

Various marching schemes can be obtained by making different choices of the operators \( P, Q \) and their discretizations \( P_n, Q_n \) and from using different methods to solve the discrete system for \( w_{n+1} \). For example, if we use central differences to approximate \( w_r \) and \( w_{rr} \), then (7) leads to a discrete system

\[
0 = Q_n^2 \left( w_{n+1} - 2w_n + w_{n-1} \right) - \frac{1}{(\Delta r)^2} \left( w_{n+1} - 2w_n + w_{n-1} \right) + 2ik_no^n \left( w_{n+1} - w_{n-1} \right) + k_0^2 (P_n^2 - Q_n^2) w_n.
\]

where \( P_n \) and \( Q_n \) are \( M \times M \) matrices.

We illustrate the above approach by specifying two such schemes.

Scheme I is the basic marching method, close to that developed in [3], obtained by making the choices

\[
P^2(h) = 1 + h, \quad Q^2(h) = 1.
\]

Then equation (8) becomes
\[
0 = \frac{w_{n+1} - 2w_n + w_{n-1}}{(\Delta r)^2} + 2ik_0(w_{n+1} - w_n) + k_0^2h_n w_n, \tag{10}
\]

where \(h_n\) is the \(M \times M\) tridiagonal matrix having \(m\)th diagonal entry \(\phi_n = (2/(k_0 \Delta x)^2)\) and all entries in the adjacent diagonals are \(a_0 = (k_0 \Delta x)^2\), except the entry in row \(M\) column \(M - 1\) is \(2\alpha_0\). Equation (10) is easily solved explicitly for \(w_{n+1}\) in terms of \(w_n\) and \(w_{n-1}\). Since \(w\) is given for \(r \leq r_0\), \(w_1\) and \(w_0\) are known and enable the marching to proceed.

Scheme II is obtained using

\[
P(h) = 1 + \frac{3}{4} h, \quad Q(h) = 1 + \frac{1}{4} h, \tag{11}
\]

so that \(P(h)\) is the usual \(\text{Padé}(1,1)\) approximation to \(\sqrt{1 + h}\). Then \(P_n\) and \(Q_n\) are 5-diagonal matrices. We use a 5-diagonal system solver in this case to solve (8) for \(w_{n+1}\).

### III. STABILITY

Each of the schemes, I and II, leads to an interpretation as

\[
W_{n+1} = M_n W_n. \tag{12}
\]

Here \(W_n = (w_n, w_{n-1})^T\) is a \(2M\) - vector and \(M_n\) is a \(2M \times 2M\) - matrix of the form

\[
M_n = \begin{bmatrix}
\eta T_n & -\tau I \\
I & 0
\end{bmatrix} \tag{13}
\]

with

\[
\eta = \frac{1}{1 + ik_0 \Delta r}, \quad \tau = \frac{1 - ik_0 \Delta r}{1 + ik_0 \Delta r} \tag{14}
\]

\[
T_n = (2 + k_0^2 (\Delta r)^2) I - k_0^2 (\Delta r)^2 Q_n^{-1} P_n^2 \tag{15}
\]

For stability, we require that all eigenvalues, \(\lambda_n\), of \(M_n\) satisfy

\[
|\lambda_n| \leq 1 \tag{16}
\]

The eigenvalues of \(M_n\) are obtained from the eigenvalues, \(z_n\), of \(T_n\) by the relation

\[
0 = \lambda^2 - \eta z_n \lambda + \tau. \tag{17}
\]

For schemes I and II the \(z_n\)’s are real and one derives from (14), (16) and (17) the stability condition

\[
x^2 < 4(1 + (k_0 \Delta r)^2). \tag{18}
\]

For both methods I and II in the constant coefficient case the \(x^2\)’s can be found explicitly and the stability condition (18) yields the following restrictions on the step sizes. These limits gave reliable guidelines for choices of step sizes in the test problems.

- Scheme I: \(\Delta x > \frac{2}{k_0^2}\) \(\Delta r < \Delta x [ (\frac{\Delta x k_0}{2})^2 - 1]^{1/2}\) \tag{19}
- Scheme II: \(\Delta x > \frac{\sqrt{3}}{k_0^2}\) \(\Delta r < \frac{1}{2k_0} [ (k_0 \Delta x)^2 - 3]^{1/2}\) \tag{20}

### IV. CALCULATIONS

We illustrate the implications and validity of the mesh restrictions (19), (20) through some test calculations performed with schemes I and II. The test problem is Part B of “test Case 2 (Bilinear Profile)” taken from the NORDA Parabolic Equation Workshop ([2], p.36). For this problem \(k_0 = 0.9\) and the most conservative estimates from (19), (20) give

- Scheme I: \(\Delta x > 22.2m\) \tag{21}
- Scheme II: \(\Delta x > 19.2m\) \tag{22}

With \(\Delta x = 25m\) in Scheme I, \(\Delta r < 12.9m\) is required by (19). With \(\Delta x = 20m\) in Scheme II, \(\Delta r < 9.95m\) is required by (20). Figure 1 shows propagation-loss curves for this problem, based on calculations using Scheme I (dashed curve) with \(\Delta x = 25m\), \(\Delta r = 1m\) and using Scheme II (solid curve) with \(\Delta x = 20m\), \(\Delta r = 3m\). The reference solution (dotted curve) was obtained using the higher order method of [5]. Note the improvement of Scheme II over Scheme I (at the cost of some small high frequency oscillation). The Scheme I curve did not improve significantly as \(\Delta x\) was decreased below 25m. In fact, the Scheme I calculation became unstable for \(\Delta x\) in the vicinity of the limiting value 22.2m given in (21).

### References


TREATMENT OF HORIZONTAL DENSITY VARIATIONS IN A 3-DIMENSIONAL OCEAN

DING LEE AND GEORGE BOTSRAS
Naval Underwater Systems Center
New London, CT 06320
U.S.A.

Abstract - A technique for handling vertical density variations in a three-dimensional ocean has been introduced by Lee-Schultz-Saad. It is generally believed that the acoustic effect of density variation in the vertical direction is stronger than that of a density variation in the horizontal direction. We extended the above treatment to handle horizontal density variations in a 3-dimensional ocean by the same numerical technique and incorporated this extended technique into the FOR3D (a 3-dimensional wave propagation prediction code). This updated code was applied to examine the effect between vertical and horizontal density variations in a selected region. Findings will be reported.

I. INTRODUCTION

Density variations in the ocean medium influence the acoustic intensity; thus, techniques must be developed to handle these density variations adequately. A number of techniques are in existence for such treatment. In this paper we enhance the numerical technique developed by Lee et al. [1] for handling vertical density variations to handle density variations horizontally. The purpose of this enhancement is to examine the effect caused by both vertical and horizontal variations. It is generally believed that the density variation in the vertical direction has a stronger influence on acoustic intensity than the density variations in the horizontal directions. A mathematical model having density variations in all directions is presented. A numerical solution follows. This numerical solution is incorporated into an existing 3-dimensional computer code FOR3D [2]. This updated code is applied to examine the overall density variations in a selected region whose inputs were obtained from Harvard University [3,4]. We report our findings based on this set of data in a selected region. Some discussions are given at the end.

II. A MATHEMATICAL MODEL AND ITS NUMERICAL SOLUTION

A 3-dimensional mathematical model [1] having vertical density variations has the expression

\[ u_r = (-ik_o + ik_x[1 + \frac{1}{2} x^2 - \frac{3}{8}(x^4)^2 + \frac{1}{2} y^2])u, \]

(1)

where

\[ x^2 = n^2(r, \theta, z) - 1 + \frac{1}{k_o^2} \rho(z) \frac{\partial}{\partial z} \left( \frac{1}{\rho(z)} \frac{\partial}{\partial z} \right), \]

(2)

\[ y = \frac{1}{k_o^2} \frac{\partial^2}{\partial \theta^2}. \]

(3)

A complete mathematical model to accommodate density variations in all directions must contain the capability of handling the horizontal density variation. A mathematical model, based upon Ref. 1, is developed to handle horizontal density variations which has the expression

\[ u_r = (-ik_o + ik_x[1 + \frac{1}{2} x^2 - \frac{3}{8}(x^4)^2 + \frac{1}{2} y^2])u, \]

(4)

where

\[ y^+ = \frac{1}{k^2} \rho(0, \theta) \frac{\partial}{\partial \theta} \left( \frac{1}{\rho(0, \theta)} \frac{\partial}{\partial \theta} \right). \]

(5)

This formulation allows Eq.(4) to be solved by the same numerical procedure used to solve Eq.(1). As a result, Eq.(4) can be solved in two steps by 2 tri-diagonal systems of equations, provided both vertical and horizontal density variations are treated satisfactorily on the interface. Vertical and horizontal density variations are accounted for in steps 1 and 2, respectively. Detailed numerical solutions of Eq.(2) as well as the numerical density treatment can be found in Ref. 1.

III. AN APPLICATION

An environment was selected from a Gulfcast generated by the Harvard Open Ocean Model [3,4]. A 25 Hz source was placed 100 meters in depth above the continental slope at latitude 39.5°N, and longitude 72°W. Propagation loss predictions were computed in a 3-dimensional wedge that measured 10° in width. Direction of propagation of the center ray emanating from the source located at the vertex of the wedge was 180°T. The water depth in the wedge is variable and ranges from approximately 500 meters to approximately 3500 meters. Density in the water and in the bottom are assumed to be 1.0 g/cm³ and 1.8 g/cm³, respectively. Four propagation loss predictions computed by the FOR3D model are shown in Fig. 1. Since FOR3D computes propagation loss in two steps, it was possible to modify the code such that different densities are accounted for in steps 1 and 2. In step 1, density variations in the vertical plane are accounted for resulting in a 2-dimensional solution. In step 2, azimuthal density variations in the horizontal plane are accounted for resulting in a 3-dimensional solution. The following table summarizes the various combinations of density conditions under which these solutions were computed.

<table>
<thead>
<tr>
<th>Curve</th>
<th>Water Density</th>
<th>Bottom Density</th>
<th>Step 1 Vertical Density</th>
<th>Step 2 Horizontal Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0 g/cm³</td>
<td>1.0 g/cm³</td>
<td>1.0 g/cm³</td>
<td>1.0 g/cm³</td>
</tr>
<tr>
<td>2</td>
<td>1.0 g/cm³</td>
<td>1.8 g/cm³</td>
<td>1.8 g/cm³</td>
<td>1.8 g/cm³</td>
</tr>
<tr>
<td>3</td>
<td>1.0 g/cm³</td>
<td>1.8 g/cm³</td>
<td>1.3 g/cm³</td>
<td>1.0 g/cm³</td>
</tr>
<tr>
<td>4</td>
<td>1.0 g/cm³</td>
<td>1.8 g/cm³</td>
<td>1.0 g/cm³</td>
<td>1.8 g/cm³</td>
</tr>
</tbody>
</table>

539
Curve 1 represents the solution at 300 meters in depth when density is set to 1.0 g/cm³ everywhere; that is, there is no density variation. For curve 2, density in the bottom was set to 1.8 g/cm³ and correctly accounted for in both steps 1 and 2. In curve 3, horizontal density variations were deliberately not accounted for; that is, no azimuthal density variation was detected by FOR3D when passing from water into a seamount. For curve 4, vertical density variations were not accounted for. A comparison of curve 1 (no density variations) with curve 2 (with density variations) shows differences of as much as 10 dB. Comparing curve 3 with 4 shows that the average level of propagation loss is midway between curves 1 and 2 and approximately the same implying that neither the vertical nor the horizontal treatment of density dominates in the solution. However, by comparing curves 1 with 4 and 2 with 3, it can be seen that the shape of the curve is influenced by the density variations in the vertical plane.

IV. CONCLUSIONS

The results show that, in general, neither the vertical nor the horizontal treatment of density variations dominate in the prediction of propagation loss. Both contribute equally well to the solution and must be accounted for. However, the shape of the propagation loss curve is determined by the density variations in the vertical plane.

ACKNOWLEDGMENT

This research was supported jointly by ONR Research Grant N00014-91-WX-24038 and Naval Underwater Systems Center Independent Research Project A65025.

REFERENCES


Abstract — The application of phase space and functional integral methods to scalar and vector, direct and inverse, wave propagation problems is briefly outlined.

I. INTRODUCTION

This paper addresses wave propagation in extended, inhomogeneous, multidimensional environments capable of channeling energy over many wavelengths. Sound propagation in the ocean and electromagnetic guided-wave propagation are two examples. For the most part, the application of classical, "macroscopic" methods has resulted in direct wave field approximations, derivations of approximate wave equations, and discrete numerical approximations. In the last several decades, however, developments in Fourier analysis, partial differential equations, mathematical physics, among others, have been synthesized into what is now called harmonic analysis in phase space [FO]. This analysis on the configuration space $R^n$ done by working in the phase space $R^n \times R^n$ has produced sharp, "microscopic" tools (pseudo differential and Fourier integral operators, wave packets) appropriate for attacking wave propagation problems in extended environments [FO]. In conjunction with the global functional integral techniques [SC] pioneered by Wiener (Brownian motion) and Feynman (quantum mechanics), and so successfully applied today in quantum field theory and statistical physics, the n-dimensional wave field propagators can be both represented explicitly and computed directly. The phase space, or "microscopic," methods and path (functional) integral representations provide the appropriate framework to extend homogeneous Fourier methods to extended inhomogeneous environments, in addition to suggesting the basis for the formulation and solution of corresponding arbitrary-dimensional nonlinear inverse problems [WE].

II. WAVE EQUATION MODELING AND FORMAL WAVE FIELD SPLITTING

For sound propagation in the ocean, the initial modeling is provided by the n-dimensional scalar Helmholtz equation,

$$\left(\nabla^2 + k^2 K^2(z)\right) \phi(z) = 0 ,$$

where $K(z)$ is the refractive index field and $k$ is a reference wave number. The environment can be characterized by a refractive index field with a compact region of arbitrary (n dimensional) variability superimposed upon a transversely inhomogeneous (n-1 dimensional) background profile. Splitting the wave field $\phi(z)$ into two components, $\phi^+(z)$ and $\phi^-(z)$, via the transformation

$$\begin{pmatrix} \phi(z) \\ \partial_z \phi(z) \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ iB_1 & iB_2 \end{pmatrix} \begin{pmatrix} \phi^+(z) \\ \phi^-(z) \end{pmatrix} ,$$

(2)

where $B_1$ and $B_2$ are the two operator solutions of the simple quadratic operator equation

$$B^2 - \left( k^2 K^2(z) + \nabla_z^2 \right) = 0 ,$$

results in the equivalent formulation [WE]

$$\partial_x \begin{pmatrix} \phi^+(z) \\ \phi^-(z) \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \partial_z B_1^{-1} & -iB_2 \\ iB_1 & \frac{1}{2} \partial_z B_2^{-1} \end{pmatrix} \begin{pmatrix} \phi^+(z) \\ \phi^-(z) \end{pmatrix} .$$

(4)

It is straightforward to show that (1) - (3) imply (4) and, conversely, that (2) - (4) imply (1). Moreover, (2) and (4) are consistent. Choosing $B_1$ to correspond to the forward (outgoing) wave radiation condition and $B_2$ to correspond to the backward wave radiation condition completes the identification.

Sound propagation in the ocean, certain guided-wave electromagnetic propagation problems, and boralche-to-boralche seismic modeling are near one-way propagation problems in extended environments. To zeroth-order, these propagation problems can be viewed as transversely inhomogeneous, suggesting a weak-backscatter perturbation approach to the general direct and inverse wave propagation problems. In this context, the transversely inhomogeneous problem is first solved and then used to attack the more general inverse problem [SC]. Mathematically, this is expressed through the approximate diagonalization of the first-order Helmholtz system in (4).

For a transversely inhomogeneous environment, $K^2(z) = K^2(z_1)$, (4) is diagonal. The forward evolution (one-way) equation,

$$(i\hbar) \partial_x \phi^+(z_1) + \left( K^2(z_1) + (1/\hbar^2) \nabla^2 \right) \phi^+ = 0 ,$$

(5)

is the formally exact wave equation for propagation in a transversely inhomogeneous half-space supplemented with appropriate outgoing wave radiation and initial-value conditions [WE].

Unlike the transversely inhomogeneous scalar Helmholtz equation, which factors in terms of a formal square root operator, the transversely inhomogeneous Maxwell's equations do not admit such a simple decomposition. The frequency-domain form of Maxwell's equations in three spatial dimensions is taken to be

$$\partial_z E(z) + \nabla \times H(z) = 0 ,$$

where, under transversely inhomogeneous conditions, the matrix operators $\Delta(z)$ and $\nabla \times$ are defined in [BR]. For transversely inhomogeneous environments, the electric field vector ($E$) can be split into physical forward ($E^+$) and backward ($E^-$) propagating wave field components. The exact diagonalization is given by [BR]

$$\partial_x \begin{pmatrix} E^+(x) \\ E^-(x) \end{pmatrix} = \begin{pmatrix} iB_1 & 0 \\ 0 & iB_2 \end{pmatrix} \begin{pmatrix} E^+(x) \\ E^-(x) \end{pmatrix} ,$$

(7)

where

$$\begin{pmatrix} \partial_x E(z) \\ \partial_z E(z) \end{pmatrix} = \begin{pmatrix} i\hbar & i\hbar \\ i\hbar & i\hbar \end{pmatrix} \begin{pmatrix} E^+(z) \\ E^-(z) \end{pmatrix} .$$

(8)

$I$ is the $3 \times 3$ identity matrix, and $B_1$ and $B_2$ are the two matrix operator solutions of the generalized quadratic operator equation

$$B^2 - i\nabla(z_1) \cdot B - \Delta^2(z_1) = 0 .$$

(9)
For a transversely inhomogeneous environment, (6) is equivalent to (7) [BR].

III. PHASE SPACE AND PATH INTEGRAL CONSTRUCTIONS

The formal one-way Helmholtz wave equation (5) can be recast and analyzed within the phase space framework as a Weyl pseudo-differential form [WE].

\[
(i/\hbar)\partial_tE^*(x,z_t) + (i/2\pi)^{n-1}\int_{R^{n-1}} dz' dp_t \exp\left(ikp_t \cdot (z_t - z_t')\right)
\]

\[
\Omega_B \left(\Psi_t, z_t, z_t'\right) = \Omega_B \left(\Psi_t, z_t, z_t', z_t''\right) \exp\left(2i\hbar(z_t'' - y - l - z_t')\right) \quad (10)
\]

In (14), \(\Omega_R(p,q)\) is defined in [BR], \(p = (p_1, p_2)\), \(q = (q_1, q_2)\), and the solution \(\Omega_B(p,q)\) corresponding to the outgoing (forward) radiation condition is chosen. The operator symbol in the scalar case has been replaced by an operator symbol matrix in the vector case. Analogous path integral and marching algorithm constructions follow [BR].

For wave propagation problems in the presence of two (generally different) transversely inhomogeneous half-spaces separated by a planar transition region of arbitrary length and inhomogeneity, the one-way phase space and path integral methods can be combined with invariant imbedding techniques. For the Helmholtz equation, designating the left and right boundaries of the transition region at \(x = a\) and \(x = b\), respectively, and generally locating a source in each half-space, the incident wave fields are connected to the scattered wave fields through the Weyl operator equation

\[
\Omega_B(p,q) = K^2(p,q) - \Delta =
\]

\[
\Omega_S(p,q) = \Omega_B(p,q) + \Omega_R(p_1, q_1) \quad (15)
\]

where \(\Omega_B(p,q)\) is defined through the Weyl composition equation

\[
\Omega_S(p,q) = \Omega_B(p,q) \cdot \Omega_R(p_1, q_1)
\]

\[
\Omega_B(p,q) = K^2(p,q) - \Delta
\]

\[
\Omega_R(p_1, q_1) = \Omega_B(p_1, q_1) \cdot \Omega_B(p_2, q_2)
\]

In (16), \(\Omega_B(p,q)\) is the operator-valued matrix in (4) evaluated at \(x = b\). The initial conditions for (16) are determined by the appropriate planar interface problem at \(b = a\). The invariant imbedding procedure transforms the Helmholtz boundary-value problem of (4) into an initial-value problem through the complete specification of the two-component wave field column vector at either \(x = a\) or \(x = b\). The formal operators in (4), (15), and (16) are explicitly represented as Weyl pseudo-differential operators [WE]. Equation (16) provides the basis for both direct and inverse algorithms [WE].

ACKNOWLEDGMENTS

This work was supported by NSF, AFOSR, and ONR.

REFERENCES


NUMERICAL SOLUTIONS OF THE ELASTIC WAVE EQUATIONS

G. H. Knightly, G.-Q. Li, D. F. St. Mary
Center for Applied Mathematics and Mathematical Computation
Department of Mathematics
University of Massachusetts
Amherst, MA 01003, USA

Ding Lee
Naval Underwater Systems Center
New London, CT 06320, USA

Abstract The question of the utilization of parabolic equations to approximate the elastic wave equations with liquid/solid interface is studied. The main consideration is the development of numerical schemes which are stable. Crank-Nicolson discretization schemes for a parabolic elastic system of equations with interface are discussed. Two approaches to the solution of the system are considered, in one the full problem is discretized directly, in the other the solid and liquid parts are solved successively.

1 Introduction

Recently, considerable attention has been given to questions revolving around the effects of the ocean bottom interface on propagation in the water column. In fact, the Office of Naval Research, USA, is currently supporting fundamental research in this area through an accelerated research initiative called Acoustics Reverberation Special Research Program. Here, the interest is in scatter from the ocean bottom, the partitioning of the incident acoustic wavefield by the bottom/subbottom, and re-radiation into the water column.

Parabolic approximation methods, having been applied extensively, and successfully, to underwater acoustics problems in the water column are beginning to be utilized in the case of an elastic medium, namely, the ocean bottom, see e.g. [6, 7, 3, 1, 2]. This activity usually involves the approximation of the elastic wave equations by a parabolic-type partial differential equation and the implementation of interface conditions to represent the liquid/solid interface under consideration. In [6], Shang & Lee choose to focus on the implementation of this interface in 2-D by working to connect the traditional parabolic equation, (PE), which is applied in the liquid medium, to a parabolic approximation of the elastic wave equations in the elastic medium. The scheme developed in [6] is unstable. In this paper, we reconsider the approach in [6], with the goal of creating a numerical scheme with robust stability properties. The essence of the approach in [6] involves the utilization of a parabolic approximation to the elastic wave equations which Shang & Lee extract from the work of McCoy [4], a clever Taylor polynomial approximation to a second derivative gleaned from [5], and three systems of equations - one representing the liquid medium, an interface system of three equations in three unknowns, and a system representing the elastic medium. In the development presented here, two approaches to the solution of the system are considered, in one the full problem is discretized directly, in the other the solid and liquid parts are solved successively. The Crank-Nicolson discretization method is employed in the case of all equations in an attempt to ensure stability.

In the next section we present the system of equations which constitutes the parabolic approximation to the elastic wave equation with interface. In section three we describe the Crank-Nicolson discretization of the system and suggest an approach to breaking it into two systems to be solved successively.

2 BACKGROUND

We adopt the notation of [6] and refer the reader to [6] for a detailed description of the derivation of the material presented in this section. The parabolic approximating equations utilized here emanate from potential equations in both media. In the elastic medium the potential equations might be thought of as a series of "local" potential equations (the medium is said to be locally isotropic) which have been joined so as to have nonconstant parameter functions. In the liquid medium the parabolic approximation process is well established and for these purposes yields the standard (PE)

\[ \frac{\partial A_1}{\partial r} = a_1 A_1 + b_1 \frac{\partial^2 A_1}{\partial z^2}, \]

where \( k_0 \) is a reference wave number. The parabolic system approximating the elastic wave equations is given by

\[ \frac{\partial A_2}{\partial r} = a_2 A_2 + b_2 \frac{\partial^2 A_2}{\partial z^2} + c_2 \frac{\partial B_2}{\partial z}, \]

\[ \frac{\partial B_2}{\partial r} = a_2 B_2 + b_2 \frac{\partial B_2}{\partial z} + c_2 \frac{\partial A_2}{\partial z}, \]

where \( a_2 \) and \( c_2 \) are coupling coefficients whose definitions along with other symbols are given by

\[ a_2 = \frac{1}{2k_2} [k_2^2(r,z) - k_0^2], \quad b_2 = i/2k_2, \]

\[ c_2 = (-1/2k_2)(\Delta k_2) \epsilon_{\alpha\nu}, \]

\[ \epsilon_{\alpha\nu} = \int_{\Delta r} r^{(n+1)/2} e_{\alpha\nu}(r,z) e^{i\Delta k_2 dz}. \]
where $\epsilon_{np} = 2(\kappa_D/\kappa_S)\epsilon_p - \epsilon_p$, $\Delta \kappa = \kappa_S - \kappa_D$, and $\ast$ denotes complex conjugate. Now

$$\kappa_D^2 = \omega^2/\omega^2_D = \kappa_S^2/(\lambda_2 + 2\mu_2),$$

and

$$\kappa_S^2 = \omega^2/\omega^2_S = \kappa_S^2/\mu_2,$$

where $\lambda$, and $\mu$ are Lamé parameters, $\omega$ frequency, and additionally, e. g., $\lambda_2$ represents an approximation to $\lambda(r, z)$ with relative error $\epsilon_3$, $\lambda_2 = \lambda [1 + \epsilon_3(r, z)],$

over an interval of length $\Delta r$.

Again, the derivation of the interface conditions in terms of the parabolic potentials is contained in [6]. We re-present them here for purposes of completeness. We would also like to call attention to the fact that in their derivation the usual association into subsystems somewhat arbitrary decisions are made, e. g. dividing the full system into two parts along the interface. Generally, the continuity of vertical components of displacement and of radial stress translate respectively to

$$\frac{\partial A_2}{\partial z} + K_D \frac{\partial A_2}{\partial z} + K_S \left( \frac{2B_2}{\partial r} + \lambda k_3 B_3 \right),$$

$$-\rho \omega^2 A_1 = 2 \mu_2 \left( \frac{\partial k_3 B_3}{\partial r} + i k_3 B_3 \right) K_S,$$

$$+ \left( \lambda_2 \left( \frac{\partial k_3 B_3}{\partial r} + i k_3 B_3 \right) - 2k_3 A_1 \right) + 2 \mu_2 \left( \frac{\partial A_1}{\partial z} \right) K_D.$$

Finally, the vanishing of the horizontal components of stress on the interface yields

$$\Delta \frac{\partial^2 A_2}{\partial z^2} = \frac{\partial^2 A_2}{\partial z^2} + 2 \kappa_D \frac{\partial A_2}{\partial z} = \left( \frac{\partial^2 B_2}{\partial z^2} - 2i \frac{\partial B_2}{\partial r} + \frac{k_3^2 B_3}{\partial r} \right) K_{SD} (2.3c),$$

where

$$K_D = \sqrt{\frac{k_0}{k_D}} e^{i \omega_D} r, K_S = \sqrt{\frac{k_0}{k_S}} e^{i \omega_S} r,$$

$$K_{SD} = K_{SD}(r) = \sqrt{\frac{k_D}{k_S}} e^{i \omega_D} r,$$

$$\Delta_D = \Delta_D - k_0, \Delta_S = \Delta_S - k_0,$$

3 DISCRETIZATION

The system of equations (2.1), (2.2), (2.3a-c) constitutes the parabolic elastic potential system with interface. We remark that each of the three equations (2.3a-c) is considered to hold on the interface. In particular, these equations represent the "slip" interface condition and thus we shall have a value for each of the unknown functions $A_1$, $A_2$, and $B_2$ on the interface. We shall perform a Crank-Nicolson discretization of each of the equations in the system. The resulting discretization scheme includes all of the unknowns in a very large system. One solution approach to this aspect of the over-all problem is to attempt to divide the problem along the natural boundary of the liquid/solid interface.

The equations (2.3a-c) require special attention since $A_1$ is defined only for values of $z$ on or above the interface, and $A_2$, $B_2$ only for values on or below it. In particular, second partial derivatives require extraordinary measures. Several possibilities suggest themselves: i) to replace the second order partial derivatives in $z$ in (2.3b-c) by using the appropriate approximating partial differential equation in (2.2) at the interface points, e. g. in (2.3b) $\partial^2 A_2/\partial z^2$ is replaced by

$$\frac{1}{b_1} \left( \frac{\partial A_2}{\partial r} - \frac{2 \partial A_2}{\partial z} - \frac{\partial B_2}{\partial z} \right)_j,$$

where $j$ represents the interface level, ii) to approximate these second order $z$-partial derivatives by a Taylor polynomial, e. g.

$$\left( \frac{\partial^2 A_2}{\partial z^2} \right)_j \approx - \frac{2}{\Delta z} \left( \frac{\partial A_2}{\partial z} \right)_j + \frac{2}{\Delta z^2} \left( (A_2)_{j+1} - (A_2)_j \right),$$

iii) to use the value of the second derivative at a lower node. It would be desirable to develop a stability analysis and/or computer implementation of each of these approaches to the "full" system. Our efforts have concentrated to date on dividing the full system into two parts along the interface. Generally, we have included the discretized (2.3a) in the liquid subsystem, and discretizations of the modified (2.3b-c) equations in the solid subsystem. In attempting to accomplish the disassociation into subsystems somewhat arbitrary decisions are made, e. g. in solving the subsystem for $A_2$, $B_2$ at the $(n + 1)$th range step, certain values of $A_1$ are required at the $(n + 1)$th range step, thus as a consequence of breaking the system into two subsystems we replace these values by the corresponding values at the $n$th range step. Methods i) and ii) have been implemented in the context just described and seem to give reasonable agreement with an exact solution of a test problem presented in [6] over the first few hundred iterations. Residuals eventually become unacceptably large as the number of iterations continue. Methods related to iii) remain a subject for future research.

References


Abstract — The improved simulation of sonic boom propagation through the real atmosphere requires greater understanding of how the transient acoustic pulses popularly termed sonic booms are affected by atmospheric turbulence. Two primary turbulence effects have been identified: (1) the thickening of the nominally abrupt shock at the beginning of the pulse waveform and (2) the spiking and rounding of the portion of the waveform that immediately follows the shock. The turbulence-induced thickening effect is typically larger than (although occasionally much weaker than) the thickening effect caused by molecular relaxation, which involves nonlinear effects. The present paper describes novel procedures to simulate these effects.

I. INTRODUCTION

Sonic boom distortion by turbulence has been considered by many authors (1–3); previous work includes attempts (generally regarded as successful) by Pierce (4) and by Crow (5, 6) to explain the random spiking and rounding of waveforms, although a complete and satisfactory statistical theory has not yet been achieved. The effects of turbulence on rise times were considered by Plotkin and George (7), Pierce (8), and by Ffowcs-Williams and Howe (9). None of the latter theories were regarded as wholly satisfactory, however, and several authors suggested that turbulence played a minor role in the rise time phenomenon and that the dominant mechanism was molecular relaxation. Recently, the author and his colleagues have discovered (10), after a somewhat careful comparison of data with theoretical predictions based on the molecular relaxation model, that that mechanism tends to underestimate actual rise times by factors of the order of three. Thus, the overall question of how turbulence affects sonic boom waveforms deserves further consideration.

II. SONIC BOOM WAVEFORMS

A typical sonic boom waveform, acoustic portion of pressure versus time, is shown in Fig. 1. Note that the peak pressure is of the order of 70 Pa and that the waveform duration...

Fig. 1. Sonic boom waveform recorded at ground during overflight of an SR-71 at 66,000 ft altitude with speed of Mach 2.6.

Fig. 2. Early portion (rise phase) of the waveform shown in Fig. 1.
is of the order of 200 ms. The standard idealization of such a waveform is that of an N-wave, an abrupt initial shock followed by a linear decrease of pressure through zero, terminated by a second abrupt shock. This particular waveform has one of the aberrations believed (4) to be caused by turbulence, in that there are small spike features near the leading and trailing edges of the shock, such that the jump in overpressure at the shocks is slightly higher than that of the overall N-shape that fits most of the waveform. Other aberrations commonly observed are rounded profiles where the initial shock is replaced by a much smaller jump followed by a slower transition of the duration of typically 10 to 15 ms up to the nominal N-shaped profile.

Figure 2 shows the first 6 ms of the waveform of Figure 1. On the scale with which this portion of the waveform is plotted, it is evident that the sudden increase in pressure corresponding to the shock is not abrupt and has a duration whose order of magnitude is 4 ms. This portion of the waveform, here termed the rise phase, is often characterized by a single number called the rise time, typically taken as that time for the pressure to rise from 10% to 90% of its peak value.

III. OUTLINE OF THE THEORY

A turbulent atmosphere may be characterized by an ambient fluid velocity $\mathbf{v}$ and sound speed $c$ that vary from point to point. Techniques for synthesizing particular realizations (drawn from an ensemble) of such fields are described in a recent paper by Karweit et al. (11). The suggestion is made here that one do the synthesis in two stages. In the first stage, one leaves out all higher order wavenumber vector components beyond some cutoff wavenumber $k_0$, to the extent one is certain that geometrical acoustics will be very nearly wholly applicable for the propagation of the sonic boom from the aircraft trajectory to the considered observation point. The resulting medium is here called the background medium.

Given such a background medium, one identifies a central ray that propagates according to the ray tracing equations described in the author's book (12) and in a recent paper (13), and one describes distance along such a ray by a parameter $s$ and nominal time of arrival (ignoring nonlinear effects) of the leading shock by $\tau(s)$. The theory originally due to Hayes (14), and described in the author's text (12), applies for nonlinear propagation in the background medium along the central ray. When such a theory is modified to take molecular relaxation into account, one has (10):

$$ p = B(s)g(s, t) \quad (1) $$

Here the amplitude function $B(s)$ continually adjusts to preserve the Blokhintzev invariant, in accord with the equation

$$ \frac{d}{ds} \left\{ \frac{B^2(s)[v + c\mathbf{n} \cdot \mathbf{A}]}{\rho c^3} \right\} = 0 \quad (2) $$

where $\delta A$ is ray tube area. The quantity $g$ satisfies the coupled equations

$$ \begin{align*}
\frac{\partial g}{\partial t} + \left( \frac{dr}{ds} \right)^{-1} \frac{dA}{ds} \frac{\partial g}{\partial s} + \frac{\partial^2 g}{\partial s^2} + \sum_\nu \left( \Delta c_\nu \right) \frac{\partial g}{\partial s} & = 0 \quad (3) \\
\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \frac{\partial g}{\partial s} + \nabla \left( \frac{\rho c}{c + \mathbf{v} \cdot \nabla} \right) \frac{\partial g}{\partial s} - \frac{\partial^2 g}{\partial s^2} & = 0 \quad (4)
\end{align*} $$

The quantity

$$ A(s) = \int_0^t \frac{\beta B(s) c_{e\text{ray}} n}{\rho (c + \mathbf{v} \cdot \nabla)} ds \quad (5) $$

is the age variable for the ray.

Let $\xi$ and $\eta$ be two arbitrarily curvilinear coordinates such that in the vicinity of the central ray, the coordinates $x$, $\eta$, and $s$, define an orthogonal curvilinear coordinate system with metric tensor equal to unity along the central ray. To discover the diffraction correction to Eq. (3), one notes that for a homogeneous medium, in the absence of dissipation, relaxation, and nonlinear effects, and with an ambient fluid velocity in only the $+s$-direction, the quantity $g$ satisfies the wave equation

$$ \begin{align*}
\frac{\partial^2 g}{\partial t^2} & + \left( \frac{\partial^2 g}{\partial \xi^2} \right) + \frac{\partial^2 g}{\partial \eta^2} = 0 \quad (6)
\end{align*} $$

For a pulse propagating nearly in the $+s$ direction, this would simplify to

$$ \left\{ \frac{\partial}{\partial t} + (v + c) \frac{\partial}{\partial s} \right\} g + \frac{2}{c} \left\{ \frac{\partial^2 g}{\partial s^2} + \frac{\partial^2 g}{\partial \eta^2} \right\} \int_0^s g ds = 0 \quad (7) $$

Consequently, the appropriate modification of Eq. (3) becomes

$$ \begin{align*}
\frac{\partial g}{\partial t} + \left( \frac{dr}{ds} \right)^{-1} \frac{dA}{ds} \frac{\partial g}{\partial s} + \frac{2}{c} \left( \frac{\partial^2 g}{\partial s^2} + \frac{\partial^2 g}{\partial \eta^2} \right) \int_0^s g ds \\
+ \frac{dA}{ds} \left( \frac{dr}{ds} \right)^{-2} \frac{\partial g}{\partial s} - \frac{\partial^2 g}{\partial s^2} + \sum_\nu \left( \Delta c_\nu \right) \frac{\partial g}{\partial s} & = 0 \quad (8)
\end{align*} $$

where $\tau(\xi, \eta, s)$ is travel time [with the total turbulence with higher wavenumbers taken into account] along paths defined by the background medium. Readers may discern some similarities in this result with the parabolic equation, especially the formulation of Kriegsmann and Larsen (15), and with the nonlinear PE formulation of McDonald and Kuperman (16).

Acknowledgements — The research reported here was supported by NASA Langley Research Center. The author is particularly grateful to Drs. Gerry McAninch and Christine Darden for their encouragement and interest.

References — Space limitations preclude explicit listings of the references cited in the text.
Abstract—The computational formula of the adiabatic modal travel time perturbation has been derived. The linear constituent and the nonlinear constituent are decomposed. Numerical results demonstrate that the significant nonlinearity of modal travel time perturbation can be caused by strong oceanic mesoscale eddies.

1. THE EXACT FORMULA

The exact modal travel time perturbation is

$$\delta t_{m} = \int_{0}^{R} \left( \frac{1}{U_m} - \frac{1}{\tilde{u}_m} \right) d\rho$$  \hspace{1cm} (1)

where $U_m$ and $\tilde{u}_m$ are modal group velocities corresponding to the unperturbed sound speed profile (SSP) $c_0(z)$ and the perturbed SSP $\hat{c}(z)$, respectively. In Ref.[1], a rigorous formulation for calculating the modal group velocity has been given:

$$\frac{1}{U_m} = c_m \int_{0}^{\infty} \frac{\psi_m}{C^2} d\zeta$$  \hspace{1cm} (2)

where $c_m$ is the modal phase velocity and $\psi_m$ is the normalized eigenfunction. Substituting eq. (2) into eq.(1), we get the exact modal travel time perturbation as follows:

$$\delta t_{m} = \int_{0}^{R} d\rho \left\{ c_m \int_{0}^{\infty} \frac{\tilde{\psi}_m}{C^2} d\zeta - c_m \int_{0}^{\infty} \frac{\psi_m}{c_0 C^2} d\zeta \right\}$$  \hspace{1cm} (3)

2. THE PERTURBATION FORMULA

The modal travel time perturbation can also be expressed by differential formulation according to the definition of modal group velocity:

$$\frac{1}{U_m} = \frac{2}{\omega c_0} \left( \hat{A}_m - \tilde{A}_m \right)$$  \hspace{1cm} (4)

Then, the modal travel time perturbation is given by

$$\delta t_{m} = \int_{0}^{R} d\rho \left[ \frac{2}{\omega c_0} \left( \hat{A}_m - \tilde{A}_m \right) \right]$$  \hspace{1cm} (5)

The integrand of eq.(5) can be obtained from the eigen-value problem. The differential equation holding for unperturbed mode is:

$$\left( -\frac{d^2}{dz^2} + \hat{c}_0^2(z) \right) \psi_m = \hat{c}_m^2 \psi_m$$  \hspace{1cm} (6)

and the differential equation holding for perturbed mode is:

$$\left( -\frac{d^2}{dz^2} + \tilde{c}_0^2(z) \right) \tilde{\psi}_m = \hat{c}_m^2 \tilde{\psi}_m$$  \hspace{1cm} (7)

Multiplying eq.(6) by $\psi_m$ and eq.(7) by $\tilde{\psi}_m$ and subtracting the products gives:

$$\left( \hat{c}_m^2 - \tilde{c}_m^2 \right) \int_{0}^{\infty} \psi_m \tilde{\psi}_m d\zeta =$$

$$\int_{0}^{\infty} \left( \hat{c}_0^2 - \tilde{c}_0^2 \right) \psi_m \tilde{\psi}_m d\zeta$$  \hspace{1cm} (8)

We take the following approximations:

$$\hat{c}_0^2 - \tilde{c}_0^2 \approx \omega_0 \left( -\frac{2 \Delta c}{c_0^2} \right)$$  \hspace{1cm} (9)

$$\left( \hat{c}_m^2 + \tilde{c}_m^2 \right) \approx 2 \hat{c}_m^2$$  \hspace{1cm} (10)

Then, we get

$$\left( \hat{c}_m^2 - \tilde{c}_m^2 \right) = \frac{-\omega_0 \int_{0}^{\infty} \psi_m \tilde{\psi}_m \frac{(\Delta c)}{c_0^2} d\zeta}{\hat{c}_m \int_{0}^{\infty} \psi_m \tilde{\psi}_m d\zeta}$$  \hspace{1cm} (11)

Substituting eq.(11) into eq.(5), the modal travel time perturbation is expressed in a "differential" form:

$$\delta t_{m} = \int_{0}^{R} d\rho \left[ \frac{2}{\omega c_0} \left( \hat{A}_m - \tilde{A}_m \right) \right]$$  \hspace{1cm} (12)
3. THE NONLINEARITY ANALYSIS

As we can see, the nonlinearity is mainly caused by the deformation of eigenfunction in terms of \( \psi_m \). By writing

\[
\tilde{\psi}_m = \psi_m + \delta \psi_m
\]  

(13)

Then, the linear and nonlinear constituent are decomposed as follows:

\[
\delta t_m = \delta t_m^{(L)} + \delta t_m^{(NL)}
\]  

(14)

where \( \delta t_m^{(L)} \) is the linear constituent:

\[
\delta t_m^{(L)} = \int_0^R dx \left\{ \frac{\partial}{\partial \alpha_m} \left[ \int_0^\alpha \left( \frac{\partial \psi_m}{\partial x_m} \right) dx \right] \right\} - \int_0^\alpha \left( \frac{\partial \psi_m}{\partial x_m} \right) dx
\]  

(15)

and \( \delta t_m^{(NL)} \) is the nonlinear constituent:

\[
\delta t_m^{(NL)} = \int_0^R dx \left\{ \frac{\partial}{\partial \alpha_m} \left[ \int_0^\alpha \left( \frac{\partial \psi_m}{\partial x_m} \right) dx \right] \right\} - \int_0^\alpha \left( \frac{\partial \psi_m}{\partial x_m} \right) dx
\]  

(16)

Consequently, the nonlinear term \( \delta t_m^{(NL)} \) vanishes when the deformation of eigenfunction \( \psi_m \) is not significant. However, there are some cases that the nonlinearity is significant and for these cases the conventional linear inversion scheme will not be appropriate. Numerical examples are presented in the following section.

4. NUMERICAL EXAMPLES

The ocean model for numerical computation consists of a canonical Munk profile [2] as background and a Gaussian eddy perturbation as follows:

\[
C_0(z) = 1500 \left\{ 1 + 0.005 \exp\left[ (z - 1000) / 1000 \right] \right\}
\]  

(17)

\[
\Delta C(r,z) = DC \cdot \exp\left\{ \frac{r - R}{D_C} - \frac{C - C_0}{D_C} \right\}^2
\]  

(18)

where \( \eta = 2(z-1000)/1000 \). The parameters of a weak cold eddy and a strong warm eddy are: DR=100 Km, DE=500 m, ZE=1000 m, DC=6 (cold), and DC=15 (warm) respectively. Modal travel time perturbations are calculated for acoustic frequency \( f=10 \) Hz and mode number \( m=1,2,3 \). The results for weak cold eddy are listed on Tab.1 and the results for strong warm eddy are listed on Tab.2.

Tab.1 Modal travel time perturbation for a weak cold eddy

<table>
<thead>
<tr>
<th>( m )</th>
<th>( t_{t_m} )</th>
<th>( t_{t_m}^{NL} )</th>
<th>( \psi_m )</th>
<th>( \psi_m^{NL} )</th>
<th>( \Delta = t_{t_m}^{NL} - t_{t_m} )</th>
<th>( Eq. 4/t_{t_m} ) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>450.1</td>
<td>448.8</td>
<td>440.5</td>
<td>-9.6</td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>369.4</td>
<td>369.1</td>
<td>341.6</td>
<td>-27.8</td>
<td>-7.5</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>257.5</td>
<td>257.5</td>
<td>245.3</td>
<td>-12.2</td>
<td>-4.7</td>
<td></td>
</tr>
</tbody>
</table>

Tab.2 Modal travel time perturbation for a strong warm eddy

<table>
<thead>
<tr>
<th>( m )</th>
<th>( t_{t_m} )</th>
<th>( t_{t_m}^{NL} )</th>
<th>( \psi_m )</th>
<th>( \psi_m^{NL} )</th>
<th>( \Delta = t_{t_m}^{NL} - t_{t_m} )</th>
<th>( Eq. 4/t_{t_m} ) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-806.3</td>
<td>-810.3</td>
<td>-1101.1</td>
<td>-294.7</td>
<td>-36.5</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>-804.8</td>
<td>-811.0</td>
<td>-853.9</td>
<td>-49.1</td>
<td>6.1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-581.7</td>
<td>-586.0</td>
<td>-613.2</td>
<td>-31.5</td>
<td>5.4</td>
<td></td>
</tr>
</tbody>
</table>

CONCLUSIONS

(1) As we can see from Tab.1 and Tab.2 that the modal travel time perturbation calculated by differential formula eq.(12) is very close to the result given by the exact formula eq.(3).

(2) The nonlinearity of modal travel time perturbation caused by a weak cold eddy \((DC=6 \text{ m/s})\), as illustrated in Tab.1, is not significant. However, for a warm strong eddy \((DC=15 \text{ m/s})\), as illustrated in Tab.2, is significant.

Acknowledgement: This work was supported by NOAA and ONR.

REFERENCES

ALGORITHMS FOR MAXIMIZING MATCHED FIELD PROCESSING OUTPUT
USED IN A NEW APPROACH TO OCEAN ACOUSTIC TOMOGRAPHY

A. Tolstoy
Acoustics Division
Naval Research Lab
Wash DC 20375 USA

L.N. Fraizer
Hawaii Institute of Geophysics
University of Hawaii
Honolulu, HI 96822 USA

Abstract

A new approach to ocean acoustic tomography uses matched field processing for narrow band, low frequency sources distributed around the region perimeter and detected on widely distributed vertical arrays. A key component to the success of this new approach is an algorithm to compute the "global maximum" of the processor outputs over the very large set of candidate environments. In this paper we discuss algorithms based on the "back propagation" algorithms used in medical tomography but modified to allow for non-uniform values along each source-receiver path and weighted according to the length of the path segment of interest. Computational results to date show that the algorithms can result in extremely accurate and efficient tomographic solutions.

I. INTRODUCTION

Ocean acoustic tomography is a technique involving the transmission of acoustic fields through an ocean region and subsequently inferring the 3-D sound-speed profiles of the region by examining those fields. Over the last decade ocean tomography experiments have shown that examination of the acoustic multipath arrivals interpreted in terms of ray theoretic models can be highly effective (Munk and Wunsch, '79; Behringer et al., '82; Cornuelle et al., '89). However, such an approach requires "high" frequency signals (above 100 Hz), and so results will be degraded by such factors as uncertainties in the source/receiver locations, internal waves and tides, rough surface scattering, etc. The measurement process itself can be extremely time-consuming requiring weeks at sea to navigate the perimeter and send signals through the region. In addition, the use of high frequencies which attenuate rapidly as a function of range limits the size of the regions which can be sampled. More generally, working with data in the time domain requires high time resolution receivers to distinguish arrivals and to detect changes in those arrivals which result from sound-speed variability. Our new technique examines interference patterns across vertical arrays of hydrophones for single frequency (not time domain) low frequency data (10 - 30 Hz) modeled by highly accurate normal mode methods. The source will be explosive shots dropped from an airplane flying around the perimeter, and so experimental time will decrease from weeks to days. The new technique will effectively transfer the burden from intense oceanographic surveys to intense computer demands.

II. APPROACH

The essence of the new approach is to find the family of sound-speed profiles which maximize the matched field processing (MFP) power1 computed for each vertical array receiving signals from the known shot sources. That is, signals received at the arrays are cross-correlated with modeled signals which have propagated through candidate environments, and we seek to maximize those correlations. If the problem is properly posed, i.e., if we impose sufficient constraints, then the maximum MFP power will occur only for the true environment.

The first stage of the process is to characterize the environment in as few parameters as possible. Oceanographers have developed a method for deriving efficient basis functions, known as empirical orthogonal functions (EOFs), from measured data (Davis, '76). Consequently, an ocean region might be very accurately described in terms of only 2 or 3 EOFs. The simulated "double eddy" environment and their associated (modified) EOFs used for the results in this paper are described in detail in Tolstoy et al., '90.

The next stage is to grid the ocean region into cells where each cell corresponds to a sound-speed profile, i.e., 2 or 3 EOF coefficients. We also need to consider the geometry of our problem: how many vertical arrays will we use and where will we deploy them; how many sources will we use and where will we drop them? For the results to be discussed here we will use four vertical arrays and 30 sources distributed as shown in Fig. 1. Each array will have 28 phones spaced at 37.5 m for processing 20 Hz signals.

The first phone will be just below the surface of the water and thus span the upper 1000 m of water where all the sound-speed variability is found. We assume that the sound-speed profiles at the source and array cells have been measured, and thus EOF coefficients are known. The complete values of the dominant EOF coefficient are shown in Fig. 2.

Finally, we need an algorithm to perform the inversion, i.e., to compute the unknown EOF coefficients which maximize the MFP power at the arrays for all the source signals.

III. THE ALGORITHM

First, we shall initialize our algorithm with a simple test environment. In particular, we will assume that we know the range of possible values for the EOF coefficients throughout our region and then use their mid-range values as a first estimate for all the unknown values (see Fig. 3) We know that this estimate is not very good because the MFP power computed at each array for each source is very low. In Fig. 4 we see a plot of the MFP power \( P_r \), for each receiver-source path for \( r = 1 \). The maximum should be about 28 (the number of phones).

Let \( \beta_1(i,j), \beta_2(i,j) \) denote the true EOF coefficients for the \( i,j \)th cell. Consider the \( i,j \)th cell, and iterate through all possible values of the EOF coefficient \( \beta_1 \). Let \( P_{r(i,j)} \) denote the maximum power found for the path from source \( s \) to array \( r \) intersecting the \( i,j \)th cell (all other coefficients along the path are fixed), i.e.,

\[ P_{r(i,j)} = \max_{\beta_1(i,j)} P_r. \]

Let \( \beta_2(i,j) \) be the corresponding coefficient, and \( \Delta^r(i,j) \) be the length of the path through the cell. Then, define the new coefficient estimate by

\[ \beta_1(i,j) = \frac{\sum_{r=1}^n \beta_{2(i,j)}(r) \Delta^r(i,j)}{\sum_{r=1}^n \Delta^r(i,j)}. \]

Next, consider \( \beta_2 \) and repeat the procedure. Then, proceed to the next cell. When all cells have been processed (one sweep), repeat from the first cell (note that all the cells may have changed their coefficients and so path contributions from the non 1st cell will have changed). For the results presented here, the process was stopped when the total power \( P_{total} = \sum P_r \) was no longer increasing.

For the example discussed, we obtained excellent results for 31 sweeps with a maximum sound-speed error everywhere of less than 0.2 m/sec. However, there were other array configurations for which the algorithm stalled, i.e., stopped before giving good results. So, we also considered variations of the algorithm by selecting the \( \beta_1(i,j) \) which simply maximized \( \sum_{r=1}^n P_r \Delta^r \), or \( \sum_{r=1}^n \beta_{2(i,j)}(r) \Delta^r \), or \( \sum_{r=1}^n P_r \Delta^r \). In general, we found that these variations sometimes improved results over the original but sometimes did not, and were prone to stalling.

CONCLUSIONS

We conclude that efficient characterization of the environment, i.e., through the use of (modified) empirical orthogonal functions, plus careful source/array geometry, can result in highly accurate estimates of the 3-D sound-speed environment. In particular, we saw that 4 vertical arrays spanning the upper 1000 m of water and placed in the center of the region of interest with shots distributed every 25 km along the perimeter resulted in maximum errors less than 0.2 m/sec for our 250 km per side square region and for the frequency 20 Hz. However, the algorithm and variations on it developed for the inverison can stall. We are presently working to find a remedy for this difficulty.

Bibliography


1See Tucker (1978) and Fric (1978) for details also of MFP.

549


1. Distribution of 4 arrays (indicated by *) and 36 sources at 100 m depth (indicated by +) for 10 by 10 grid covering 250 by 250 square km.

2. Plot of the dominant EOF coefficient $\beta_i(j)$ as a function of range and cross-range. The scale has been normalised so that negative values range from -1 to 0, positive values from 0 to +1.

3. Initial estimates for EOF coefficient $\beta_i(j)$ as a function of range and cross-range. The coefficients around the perimeter (where the sources are located) and at the arrays are known; otherwise, $\beta_i = $ constant.

4. Initial estimates for MFP power $P_s$ at array $r = 1$ for each source $s$ given initial environment of Fig. 3.

5. Final estimates for EOF coefficient $\beta_i(u)$ as a function of range and cross-range. Compare to Fig. 2.
Abstract - Seismic forward modelling is one of the foremost tools for investigating wave propagation in complex geological structures. Moreover, due to the complexity, both lithological and stratigraphical, that can be found in such structures, the use of a numerical method with great accuracy and flexibility is needed for correct results. In this respect, the Spectral Element Method (SPEM) which combines the accuracy of spectral techniques and the flexibility of finite element methods is well suited. The method has been applied to solve the acoustic wave equation. Accuracy and convergence properties of Chebyshev SPEM are discussed in the present work.

1. INTRODUCTION

The most widespread of the discrete numerical methods for modelling seismic wave propagation are the finite difference (FDM) [1],[6], the Fourier or pseudo spectral (SPH) [4],[7],[10], and the finite element (FEM) methods [9],[10]. Even though based on different mathematical approaches, all three methods rely on the space discretization of the geological structure to be modelled. In particular, the pseudo spectral method can be seen as a limiting case of the finite difference spectral methods of increasing order and accuracy [3]. The main advantage of the FSM is its great accuracy, allowing for a lower number of grid points per minimum wavelength propagating in the model; a saving up to several orders of magnitude in computer memory and time can be realized. On the other hand, the FEM is well known for its flexibility in describing problems with complex geometries; irregular surfaces between different media can be defined with great accuracy. A method which combines FSM accuracy with FEM flexibility is desirable for seismic wave modelling. A high-order finite element spectral method [12],[13], seems a good candidate for this.

In the present work, we shortly present the Chebyshev spectral element method and then we discuss the implementation and the numerical results obtained for the one dimensional acoustic wave equation. The accuracy and convergence properties of SPEM are investigated by comparison with standard finite element method and with an analytical solution. Estimations are carried out by computing a frequency error index function, that relates numerical and analytical results in the frequency domain, for interpolants of increasing order.

II. CHEBYSHEV SPECTRAL ELEMENT METHOD

The Fourier spectral technique is a particular case of the more general spectral methods (SPH) [5], and both spectral and finite element methods can be seen as particular cases of the class of discrete numerical techniques for solving differential equations known as the method of weighted residuals (MWR). With the MWR, the solution is obtained by minimizing the residual i.e., the error in the differential equation produced by using a truncated expansion instead of the exact solution, with respect to a suitable norm. To this end, a set of trial functions and a set of weight functions must be defined. The trial functions are used as the basis functions for the truncated series expansion of the solution; the weight functions are used to ensure that the differential equation is satisfied as closely as possible by the truncated series expansion. The choice of trial functions is different for the two methods. In the case of the spectral methods, the trial functions are infinitely differentiable global functions, while for the finite element methods, the domain is divided into small elements, and a trial function is specified in each element. In the latter case, the trial functions are with local support, and so well suited for handling complex geometries. Following the Galerkin approach, the weight functions are the same as the trial functions and are, therefore, smooth functions which individually satisfy the boundary conditions.

The Chebyshev SPEM that we present in this part of the paper is based on the idea of decomposing the spatial domain, where the physical problem must be solved, into subdomains, as in FEM, and then on each subdomain expressing the solution of the problem we are looking for by a truncated expansion of orthogonal polynomials, as in SPH. More specifically, in the case of one dimensional problems, we decompose the original spatial domain \( \Omega \) into non overlapping elements \( \Omega_c \), where \( c = 1, ..., n_e \), and \( n_e \) is the total number of elements. As approximating functions on each element \( \Omega_c \), we chose functions belonging to \( \mathbb{P}_{n_e} \), space i.e., polynomials of degree \( \leq n_e \) in \( x \). Then the global approximating function is build up as a sum of the elemental approximating functions and, therefore, is a continuous function which is a piecewise polynomial defined on the decomposition \( \hat{\Omega} \) of the original domain \( \Omega \). The continuity of the derivative at the element interfaces is not satisfied for fixed \( n_e \), but only as a consequence of the convergence process i.e., when all \( n_e \) tend to infinity.

We now discuss the construction of such
approximating functions using Chebyshev orthogonal polynomials. A function \( f(x) \), defined on the interval \([-1, 1]\), can be approximated by a truncated expansion of Chebyshev polynomials as follows:

\[
f(x) \approx I_N(f(x)) = \sum_{k=0}^{N} \hat{c}_k T_k(x),
\]

where \( T_k \) are the Chebyshev polynomials defined as

\[
T_k(\cos \theta) = \cos k\theta
\]
or, equivalently, with the recurrence relation

\[
T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x) \quad \forall k \geq 1,
\]

and \( T_0(x) = 1 \), \( T_1(x) = x \). Using the orthogonality property of \( T_k \) and the Gauss-Lobatto quadrature formula, the expansion coefficients \( \hat{c}_k \) are easily computed. It follows that the interpolant of \( f \) can be written as

\[
I_N(f(x)) = \sum_{j=0}^{N} f(x_j)\phi_j(x),
\]

where \( \phi_j(x) \in \mathcal{P}_N \) are Lagrangian interpolants satisfying the relation \( \phi_j(x_k) = \delta_{jk} \) within the interval \([-1, 1]\) and identically zero outside. The Lagrangian interpolants are given by

\[
\phi_j(x) = \frac{2}{N} \sum_{k=0}^{N} \frac{1}{\bar{c}_j \bar{c}_k} T_k(x_j)T_k(x),
\]

with \( \bar{c}_j = \{1 \text{ for } j \neq 0, N \} \), \( \bar{c}_0 = 2 \), and where \( x_j \) are the Chebyshev-Gauss-Lobatto quadrature points

\[
x_j = \cos \left( \frac{\pi j}{N} \right)
\]

for \( j = 0, \ldots, N \).

In order to apply these interpolants to the spatial decomposition, we define the mapping \( F(\xi(x)) = \xi(x) \in \Omega_0 \rightarrow (\xi(x) \in [-1, 1]) \), between the points \( x \in [a_0, a_0+1] \) of the element \( \Omega_0 \) and the local element coordinate system, by

\[
\xi(\theta) = (x-a_0) \in \left[0, \frac{1}{\Delta_e} \right],
\]

with \( \Delta_e = a_{e+1} - a_e \). The Chebyshev-Gauss-Lobatto interpolants in the \( \Omega_0 \) element are then written as

\[
\phi_j(\xi(\theta)) = \frac{2}{N} \sum_{k=0}^{N} \frac{1}{\bar{c}_j \bar{c}_k} T_k(\xi(\theta))T_k(\xi(\theta)),
\]

where \( \xi(\theta) \) are the Gauss-Lobatto points in the local coordinate system.

For the spatial discretization of a two-dimensional problem, the Cartesian products of the Chebyshev-Gauss-Lobatto points are used on each rectangular element \( \Omega_0 \) and the Lagrangian interpolants are represented using tensor products of Chebyshev polynomials.

### III. ONE DIMENSIONAL WAVE EQUATION

To illustrate SPERM in practice, we use as model problem the one-dimensional wave equation which describes the propagation of longitudinal waves in an elastic rod. The initial boundary value problem for a rod of length \( L \) and with fixed boundaries can be stated as follows:

\[
given u_0 \text{ and } \dot{u}_0, \text{ find a continuous function } u(t) \in \Omega \rightarrow \mathbb{R} \text{ such that it satisfies the equation}
\]

\[
\frac{1}{c^2} \frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} = 0 \quad \text{on } \Omega \times (0, T),
\]

\[
with u(0, t) = u(L, t) = 0 \quad \forall t \in (0, T), \\
and \dot{u}(x, 0) = u_0(x), \quad \ddot{u}(x, 0) = \ddot{u}_0(x) \quad \forall x \in \Omega,
\]

where the dot in \( \dot{u} \) indicates partial differentiation with respect to time and where \( u(x, t) \) is the axial displacement, \( c^2 = E/\rho \) is the characteristic velocity (velocity of sound), \( E \) is Young's modulus, \( \nu \) is the rod density, and \( \Omega = [0, L] \).

To illustrate the SPERM approach, we consider the one-dimensional case. Using these approximating function spaces and the mapping \( F(\xi(x)) \) to solve equation (9), by straightforward computation, it can be shown that the one-dimensional wave propagation problem becomes as follows:

\[\text{given } u_0 \text{ and } \dot{u}_0, \text{ find } \hat{u} \in S_1 \text{ such that for all } \dot{u} \in V_1 \text{ the following equations are satisfied:}\]

\[
\phi_j(\xi(\theta)) = \sum_{j=0}^{N} \phi_j(\xi(\theta)) \phi_j(\xi(\theta)),
\]

\[
\text{with } \hat{u}(\xi(\theta)) = \hat{u}_0(\xi(\theta)), \quad \phi_j(\xi(\theta)) = \phi_j(\xi(\theta)),
\]

where \( \hat{u}_0(\xi(\theta)) = \hat{u}_0(\xi(\theta)), \) and \( \phi_j(\xi(\theta)) = \phi_j(\xi(\theta)) \) are the grid values of the unknown approximate solution and of the weight functions, respectively. Using these approximating function spaces and the mapping \( F(\xi(x)) \) to solve equation (9), by straightforward computation, it can be shown that the one-dimensional wave propagation problem becomes as follows:

\[
given u_0 \text{ and } \dot{u}_0, \text{ find } \hat{u} \in S_1 \text{ such that for all } \dot{u} \in V_1 \text{ the following equations are satisfied:}\]
\begin{align}
\sum_{e=1}^{n_e} \left[ \frac{d^2}{dt^2} \ddot{\mathbf{u}}_e - \frac{1}{c^2} \mathbf{u}_e \right] + \mathbf{a}(\mathbf{u}_e, \mathbf{\dot{u}}_e) = 0 ,
\end{align}

where \( \mathbf{a}(\mathbf{u}_e, \mathbf{\dot{u}}_e) \) is the bilinear form given by

\begin{align}
\mathbf{a}(\mathbf{u}_e, \mathbf{\dot{u}}_e) &= \sum_{i=0}^{n_e} \sum_{j=0}^{n_e} \mathbf{m}_{ij}^{(e)} \mathbf{\dot{u}}_i(t)^T \mathbf{\dot{u}}_j(t) ,
\end{align}

with \( \mathbf{m}_{ij}^{(e)} \) being the elemental mass and stiffness matrices,

\begin{align}
\mathbf{m}_{ij}^{(e)} &= \frac{2\Delta_t}{c^2(N_0)^2} \sum_{p=0}^{N_0} \sum_{q=0}^{N_0} \mathbf{p}_{pq} T_p \mathbf{t}_q \mathbf{t}_q^T \mathbf{t}_p \mathbf{t}_p^T \mathbf{m}_{pq} ,
\end{align}

and

\begin{align}
\mathbf{k}_{ij}^{(e)} &= \frac{8\Delta_t}{(N_0)^2} \sum_{p=0}^{N_0} \sum_{q=0}^{N_0} \mathbf{p}_{pq} T_p \mathbf{t}_q \mathbf{t}_q^T \mathbf{t}_p \mathbf{t}_p^T \mathbf{k}_{pq} ,
\end{align}

where \( \mathbf{p}_{pq} \) is the connectivity matrix defined in [11].

Let us define the connectivity matrix \( \mathbf{p}(t) \) such that it satisfies the discrete solution \( \ddot{\mathbf{u}}(t) \) for all Chebyshev points \( \mathbf{X}(j) \), for \( j=0, \ldots, N_p \), except for \( j=0, \ldots, N_g \) except for \( \mathbf{X}(j)^{[p]}=0 \) and \( \mathbf{X}(j)^{[p]}=1 \).

The matrices \( \mathbf{M} \) and \( \mathbf{K} \) are positive-definite, symmetric, and band-limited, the bandwidth being determined by the largest \( N_g \). They can be easily computed for each chosen order of the interpolants.

To solve the system of linear, second order, ordinary differential equations with constant coefficients just derived, we must integrate over the time interval \([0, \tau]\). This is done by discretizing the time variable as \( t_n = n\Delta_t \), \( 0 \leq n \leq N_T \), where \( \Delta_t = \tau/N_T \), and \( N_T \) is the total number of time steps. At time \( t_n \), the solution will be \( \mathbf{u}_n = \mathbf{U}(t_n) \). From the different time integration schemes which are available, we used the Newmark central difference scheme which is an implicit two-step scheme, conditionally stable and second order accurate.

IV. ANALYSIS AND DISCUSSION OF SPEN

In order to investigate the accuracy of the method, comparisons were done between the analytical and numerical solutions of the one dimensional problem expressed by equation (10). The general solution of the problem is given by
where $\hat{u}_0$ and $\hat{d}u_0/\hat{t}$ are the odd 2L periodic extensions of $u_0$ and $\dot{u}_0$ to the entire real axis [2],[14].

Discrete numerical methods for solving a PDE introduce errors in the sought solution but they get progressively less as the mesh size becomes finer. Numerical modelling for wave propagation actually behaves as a low pass filter in the sense that low frequencies accurately propagate through the mesh whereas high frequencies are undesirably modified [10]. The most evident numerical effects (in the high-frequency band) are numerical attenuation, numerical anisotropy, dispersion in numerical phase and group velocity, and numerical polarization. Due to this fact, frequency analysis is a very suitable approach for the investigation of numerical modelling results. It makes it easier to determine the spectral band in which the error is solved correctly; that is to find the minimum wavelength for which a given numerical method is accurate. Following this approach, we introduce the "frequency error-index", a complex function defined as

$$H(\omega) = \frac{\hat{u}_{\text{num}}(\omega)}{\hat{u}_{\text{an}}(\omega)} = |H(\omega)| e^{i\phi(\omega)},$$

where $\hat{u}_{\text{an}}(\omega)$ and $\hat{u}_{\text{num}}(\omega)$ are, respectively, the Fourier transforms of the analytical and numerical solutions that we want to compare. The frequency error-index makes the interpretation of the results effective since it is not affected by the spectrum of the chosen initial impulse but depends only on the numerical model. By considering both the real and imaginary part of the frequency error-index, the decay of accuracy due not only to amplitude variations (when $|H(\omega)| \neq 1$) but also to velocity dispersion (when $\phi = \arctg \left( \frac{\text{Im}[H(\omega)]}{\text{Re}[H(\omega)]} \right) \neq 0$) can be investigated.

Moreover, in order to get an idea of the rate of accuracy decay in time for each polynomial order, we can use the frequency error-index function. It is clearly important that, for large scale modelling in particular, not only efficiency (the number of grid-points per wavelength) but also accuracy of a numerical method be stable in time (after a certain reference distance of propagation).

We discuss now the numerical experiments performed and the results obtained. Let us denote the number of grid-points per wavelength by $G$. The following relations hold: $1/G = \Delta x/\lambda$, $0 < 1/G \leq 0.5$. As initial condition for the displacements, this function was chosen such that

$$u(x, t) = A \cos(2\pi x G) \exp[B(x - x_0)^2].$$

Analytical and numerical solutions were compared at travel distances in which the analytical solution reasserts the original form (because of the periodicity of the solution (28)). Results of simulations by FEM with polynomial orders $N = 1, 2, 3, 4, \text{and \ SPH \ with \ orders \ } N = 5, 8, 12, 15, 20, 30, 40, 60 \text{ \ have \ been \ collected. \ No \ orders \ less \ than \ } N=5 \text{ \ were \ chosen \ for \ SPH \ because \ of \ the \ fact \ that \ the \ Chebyshev \ collocation-points \ for \ low \ polynomial \ order \ are \ very \ close \ to \ the \ equispaced \ points, \ so \ that \ no \ difference \ between \ the \ solutions \ obtained \ with \ the \ two \ methods \ can \ be \ expected \ in \ practice. \ During \ the \ experiments, \ the \ number \ of \ nodes \ was \ held \ constant (N = 120), \ and \ time \ steps \ were \ chosen \ for \ each \ polynomial \ order \ such \ as \ to \ ensure \ numerical \ stability. \ The \ first \ set \ of \ experiments \ consisted \ in \ simulating \ the \ propagation \ for \ 2800 \ grid \ points \ of \ a \ broad-band \ impulse \ through \ the \ elastic \ homogeneous \ rod, \ both \ with \ FEM \ and \ SPH, \ but \ with \ different \ polynomial \ orders. \ By \ using \ a \ broad-band \ impulse, \ we \ shall \ see \ that \ it \ is \ possible \ to \ identify, \ for \ each \ order \ of \ approximation, \ a \ low \ frequency \ band \ where \ the \ analytical \ and \ the \ numerical \ solutions \ agree. \ The \ maximum \ of \ this \ interval \ gives, \ for \ the \ order \ $N=5$, \ the \ minimum \ $G_\text{min}(N)$ corresponding to the maximum of this interval gives a wavelength $\lambda_\text{min}(N)$ that is the minimum wavelength that the model can propagate without appreciable errors. \ In \ a \ second \ set \ of \ experiments, \ we \ used \ an \ impulse \ with \ a \ spectral \ band \ appropriately \ bounded \ in \ high \ frequency \ in \ order \ to \ prove \ that, \ in \ this \ case, \ the \ propagation \ errors \ are \ negligible. \ As \ an \ example, \ figures \ 1 \ and \ 2 \ show \ the \ frequency \ error-index \ in \ both \ amplitude \ and \ phase, \ respectively, \ for \ SPH \ with \ polynomial \ order \ $N=15$ \ and \ for \ a \ propagation \ of \ 960 \ grid-points \ for \ a \ broad-band \ impulse, \ where \ analytical \ and \ numerical \ spectrums \ are \ in \ good \ agreement, \ is \ well \ defined \ both \ by \ the \ amplitude \ and \ phase \ of \ the \ frequency \ error-index. \ The \ phase, \ however, \ shows \ very \ high \ instability \ just \ after \ the \ value \ of \ $G_\text{min}$. \ From \ this \ curve \ it \ can \ be \ inferred \ that \ in \ the \ low-frequency \ band \ (that \ is \ for \ $G < G_\text{min}(\Delta x)$) the \ model \ is \ very \ stable, \ while \ for \ $G > G_\text{min}(\Delta x)$ \ the \ model \ becomes \ unstable. \ Similar \ results \ have \ been \ obtained \ in \ all \ the \ other \ cases. \ Thus \ no \ numerical \ dispersion \ is \ observed \ for \ waves \ which \ have \ a \ frequency \ in \ the \ correct \ range.
maximum. It should be noted that according to the "dx max" estimate, the asymptotic value is \( G \approx 3 \), but this corresponds to \( G \approx 2 \) according to the "dx max" estimate. That is, the theoretical limit of the spectral methods, \( G = 2 \), is reached locally with SPH, which corresponds globally to \( G \approx 3 \) for the Chebyshev mesh.

Figures 4 shows the amplitude of the frequency error-index for the propagation of a pulse whose spectrum is inside the model frequency band (defined by \( \omega < 1/(G(N)\Delta x) = \omega_N \)) for order \( N=15 \) (SPH); for this order, \( G_{\text{min}}(\omega) = 6 \). For a propagation of 960 grid-points, the computed spectrum has in practice negligible errors, and the cut-off frequency is reached abruptly with almost no errors in the high part of the model spectrum band. The wave-forms at the same travel distance are shown in Figures 5 and 6 using a broad-band impulse for a 3-order FEM and a 15-order SPH, and in Figure 7 using the band-limited impulse, respectively. In the last case, the errors are of the order of \( 10^{-4} \). From a comparison, it is evident that SPH is globally better performing and almost non-dispersive, as expected.

As a final point, \( G \) values were collected at different time steps and for different polynomial orders. In Figure 8, curves are drawn representing the variation of \( G \) with travel distance for polynomial orders 1, 2, 3 (FEM), 4 (FEM and SPH), and 15, 50 (SPH). For low order FEM the curves show a quite high rate of increase, thus a large number of grid points per wavelength must be chosen for long
The main results can be summarized as follows:

- low values of $G$, the number of grid points per wavelength, can be achieved (close to the theoretical minimum);
- great accuracy for the propagation of waves whose spectrum lies below the wavelength corresponding to the correct $G_{\min(N)}$ is obtained;
- high-order schemes are very stable in time with respect to $G$ and accuracy; they show almost no dispersion;
- the order of the interpolants can be changed very easily.

V. CONCLUSIONS

In this work the Chebyshev spectral element method has been presented. The SPEM solution of the one dimensional wave equation has been illustrated to demonstrate the feasibility of the method for wave equation problems. SPEM accuracy and convergence properties have been investigated numerically by computing the frequency error-index function for a large set of numerical experiments. As a conclusion, for the high-order Chebyshev spectral element method applied to the acoustic wave equation, the main results can be summarized as follows:

- low values of $G$, the number of grid points per wavelength, can be achieved (close to the theoretical minimum);
- great accuracy for the propagation of waves whose spectrum lies below the wavelength corresponding to the correct $G_{\min(N)}$ is obtained;
- high-order schemes are very stable in time with respect to $G$ and accuracy; they show almost no dispersion.

REFERENCES


Abestract - We consider CW radiation by a point source in the presence of a two-scale randomly-layered medium (a slab of finite thickness or a random half space). The source can be exterior to or buried within the random layer. Expressions are obtained for the first moment and two-point correlation function of the acoustic pressure.

Introduction.

Abyssal plains comprise much of the world's ocean bottom and consist of a sediment layer (nominally one kilometer thick) lying upon a rock basement. The sediment layer is itself of heterogeneous composition and possesses acoustic constitutive parameters (density and sound speed) that vary on two length scales [1]. On one hand, the formative deposition processes have created a material, layered with clay, silt, pelagic remains and the like, whose constitutive parameters fluctuate rapidly with depth (on the scale of centimeters). On the other hand, compactification formation of the transverse spatial coordinates leads to the following system of stochastic ordinary differential equations.

\[ \frac{d\rho}{dz} = \frac{i\omega}{c} (K^{-1} - \rho^{-1}K^2)\rho + \rho_0^2 \delta(z - z_0) K^{-1} \]

\[ \frac{d\rho_a}{dz} = \frac{i\omega}{c} (\rho(z)\delta(z - z_0)) \]

where \( \delta(z) \) is the delta function.

Some Results.

Using the asymptotic theory derived in [3], we can characterize the first and second moments of the reflected and transmitted pressure. For brevity, we here give results for the simplest configuration, for the pressure reflected from a lossless random half space \( \rho = 0, \rho_0 = \rho_p \) having a constant deterministic background and only random sound speed fluctuations. For this case, the coherent reflected pressure is:

\[ E[p_{ref}(x,y,0,t)] = \frac{-i\omega}{c} [x_1 + y_2 - x_2] \left[ \begin{array}{c} x_1 + y_2 - z_1 \\ \ \ 0 \end{array} \right] \]

\[ \left[ \begin{array}{c} 0 \\ \ \ 0 \end{array} \right] \left[ \begin{array}{c} x_1 + y_2 - z_1 \\ \ \ 0 \end{array} \right] \]

\[ \left[ \begin{array}{c} 0 \\ \ \ 0 \end{array} \right] \left[ \begin{array}{c} x_1 + y_2 - z_1 \\ \ \ 0 \end{array} \right] \]

where \( R \equiv x^2 + y^2 \), \( z_1 \equiv (K_i\rho)^{1/2} \), \( j = 1 \).

Thus the coherent reflected pressure is determined by reflection from an effective medium, characterized by \( \rho_p \) and \( \rho_p \). When the problem is further simplified by assuming that the medium is matched (i.e., \( \rho_p = \rho_p \)) and the source is placed on the interface \( \rho = 0 \), the reflected pressure intensity becomes:

\[ \mathcal{E}(|p_{ref}(x,y,0,t)|^2) = \frac{\omega^2}{32\pi^4} \int_0^1 d\gamma y^2 \left[ y^2 - \gamma^2 + \frac{\gamma^2}{2} \right]^2 \]

\[ \left[ \begin{array}{c} x_1 + y_2 - z_1 \\ \ \ 0 \end{array} \right] \left[ \begin{array}{c} 0 \\ \ \ 0 \end{array} \right] \left[ \begin{array}{c} x_1 + y_2 - z_1 \\ \ \ 0 \end{array} \right] \]
where the localization length $l$ is given by

$$l = 2c^2 \left[ \omega^2 \int_0^{\infty} \lambda \, \lambda^2 \, \lambda^2 \right]^{-1}$$  \hspace{1cm} (5b)$$

Note that when radial distance from the origin is much greater than a localization length, i.e. when $\frac{r}{2l} >> 1$, the reflected intensity becomes approximately equal to:

$$E(p_{\text{refl}}(x,y,0)^2) \approx \frac{\omega^2}{128 \pi c^4 l^3}$$  \hspace{1cm} (6)$$

Thus, the theory predicts that a substantial amount of acoustic energy is reflected from the randomly layered medium.

References,


ACTIVE TIME DELAY AND PHASIS ESTIMATION IN UNDERWATER ACOUSTICS
G. JOURDAIN
CEPHAG/IEG BP 46, 38402 St Martin d'Hères, France.

Abstract

We are interested in active identification of the multipath underwater (u.w) propagation, ie in the estimation of path parameters (delay, amplitudes and phases). This completes the prediction given by the wave propagation equation. The classical method for identifying the u.w channel response is presented, and some examples of u.w multipath channels are given. The problem of path parameter estimation becomes more difficult when the paths are very close. We have proposed some high resolution methods - or joint estimation methods - which enable to solve close paths, estimate their parameters and follow their time variations. Some results applied to u.w data are finally given.

I - MULTIPATH PROPAGATION IN U.W ACOUSTICS

It is now well admitted the u.w channel can be modelled as a linear filter \( \mathcal{F} \) between an emitter E and a receiver R, moreover the additive noise.

\[
\begin{align*}
  s(t) & \rightarrow \mathcal{F} \rightarrow x(t) \rightarrow \mathcal{F}^{-1} \rightarrow b(t) \\
  E & \rightarrow & R
\end{align*}
\]

(1)

Fig 1 : The u.w channel model

This filter takes account of the energetic aspect (absorption, diffusion...) but also of temporal and frequencial distortions, and particularly multipath effects.

According to the propagation and geophysical conditions (E, R, bottom...), to the carrier frequency \( \nu_0 \), and also the time scale of interest, this filter can be modelled either random, or deterministic, or time varying [1]. Anyway the description of \( \mathcal{F} \), and its time or stochastic variations, is a main complement to the prediction given by the solution of sound propagation equation in u.w acoustics (ray or mode theory, or hyperbolic equation...) which only gives an approximate and static solution.

As the emitted and received signals are always band pass around \( \nu_0 \), we use the complex amplitudes relative to \( \nu_0 \), denoted \( \alpha_k \) for emission, \( \gamma_k \) for reception, \( H(t) \) the corresponding channel band pass impulse response (1.d). The multipath propagation is traduced by

\[
y(t) = \sum_{k=1}^{p} \alpha_k e^{i\phi_k} c(t-t_k) + n(t)
\]

(2)

\[
H(t) = \sum_{k=1}^{p} \alpha_k e^{i\phi_k} \delta(t-t_k)
\]

(3)

where the parameters \( \alpha_k, \phi_k, t_k \) are assumed here constant, but they can be random, time varying... \( \alpha_k \) traduces the energetic level transmitted over the delay \( t_k \), the phase \( \phi_k \) includes the phase relative to the path \( k \) plus a delay term (2 \( \pi \nu_0 t_k \)) plus an eventual demodulation phasis.

II - ESTIMATION OF THE IR EXPERIMENTAL EXAMPLES

The problem of interest in this paper is the estimation of the u.w channel. Let us note we perform here active identification, and not passive time delay estimation (see for ex [2]). It has been shown [3] the minimum output error solution for the channel identification is to emit large WT product (W bandwidth, T duration) signals, and to cross correlate the channel output with a copy of emission in baseband notation this leads to

\[
\Gamma_{e_\nu}(t) = \sum_{k=1}^{p} \alpha_k e^{i\phi_k} \Gamma_c(t-t_k) + b(t)
\]

(4)

where \( \Gamma_{e_\nu} \) is the auto correlation function and \( b(t) = \Gamma_{y_n} \).

Two examples are given below, coming from 2 sea experiments. The first (fig 2a) has been obtained in the following conditions : shallow water ; E/R distance = 4 km; \( \nu_0 \sim 500 \text{ Hz}. \) The 2nd example is deep water, and more stable because \( \nu_0 = 60 \text{ Hz} \). \( d_{E'R} = 140 \text{ km}. \) The multipath structure is evident on both figures with different typical time variations of parameters.

III - PARAMETER ESTIMATION OF THE MODEL (3)

III-1. One path case : In this case (p = 1), and if \( n(t) \) is white, gaussian, with psd \( \gamma_0 \), the above mentioned cross correlation (CCOR) leads to the Maximum Likelihood (ML) estimates of \( \alpha, \phi \), denoted \( \hat{\alpha}, \hat{\phi} \), introduced by \( \hat{\tau} \), \( \hat{\delta} \), \( \hat{\xi} \). \( \hat{\gamma} \) is the delay for which this CCOR is maximum, \( \hat{\delta} \) and \( \hat{\xi} \) are the modulus and phasis of the maximum of CCOR. It is well known these estimates are asymptotically unbiased and their variance is bounded by Cramer Rao bound (the performance is directly connected to SNR and \( \hat{\gamma} \) effective signal bandwidth - see [3]). As soon as \( p \geq 2 \), the calculus of the structure of joint ML estimate of the parameter set becomes complex, and the performance calculus too, particularly when the paths are very close (closer than 1/T), ie when they are no longer distinguished by the CCOR processing. (Let us note the cases of 2 and 3 joint paths have been treated [4]). So in the examples of fig 4, some of the paths are well separated, whereas some other are indistinguishable.

III-2. Close paths estimation. Different solutions have been proposed in order to solve and estimate close paths after the CCOR step. This first step is important because SNR is improved and some paths are already resolved. A first kind of methods (see [5] and references) consists in transposing the high resolution (HR) methods well known in spatial or frequencial filtering (Musique methods...). We have used the Tufts Kumaresan method for the case of fig 2b [5]. Without detailing this method, let us say i) it needs a deconvolution step, ii) the estimation of the amplitudes \( \alpha_k e^{i\phi_k} \) is uncoupled of \( t_k \) estimation, which is not optimal.
In any case the number p of paths must be first estimated. Let us see now a second kind of method which directly joint estimates the set of parameters of the model (3).

IV - JOINT ESTIMATION OF DELAYS, PHASES AND AMPLITUDES

IV 1 The method: By time sampling the equation (4), the following matrix/vector equation is obtained

\[ \Xi = M(\tau, \Theta) \alpha + b \]  

where \( M(\tau, \Theta) \) is the model matrix (where \( \tau \) is unknown, but \( \Theta \) is known) and \( \alpha \) is the unknown complex amplitude vector of \( \alpha_k e^{j \phi_k} \).

The minimization of

\[ J_\alpha = \| \Xi - M \alpha \|_2^2 \]  

versus the \( \alpha \) and \( \Theta \) parameters is the optimal least square solution (or ML solution if \( b \) is gaussian). In this case too, the number p must be first estimated by a detection criterion for example; practically one can also test several values of \( p \). These optimal estimates are (4):

\[ \hat{\tau} = \text{Arg min} \| P - y \|_2^2 \]  

\[ \hat{\Theta} = \text{Arg min} J \]  

where \( P = I - M(M^TM)^{-1}M^T \)

The minimization of \( J \) is performed by a gradient descent algorithm judiciously initialized (4).

IV 2 Sequential estimation: The above minimization of \( J \) is applied to each received data \( \Xi_n \) and each CCOR \( \gamma_{yc}(t) \). In our u.w. case, as shown in fig 2, one always disposes of several successive \( \gamma_{yc}(t) \). We have recently proposed (4) to improve the procedure by taking account of successive intercorrelations. The above non linear estimation method enables to easy introduce this as a priori information. This is equivalent of adding a constraint equation on some parameter \( \beta \) in (6) ("regularization"). So now one tries to minimize for each record \( \gamma_{yc} \)

\[ J_n = \| \Xi_n - M(\tau_n, \Theta_n) \alpha_n \|_2 + \mu \| \Theta_n - \Theta_n - 1 \|_2 \]  

In the u.w case, as the delay positions \( \tau \) are stable enough from one record to another, we use only \( \Theta \) as constraint parameter \( \Theta \) in (6).

IV 3 Results: In the fig 2b, there are 3 "main paths" but there are perhaps some unsolved paths inside them and the ray tracing predicts 2 close paths inside the second "path". First a zoom is made of this second "path" (fig 3a). One tries to identify a model like (5) in it.

The results are given in fig 3b. \( \mu = 0 \) corresponds to the minimization of \( J \) (6), and in fig 3c, \( \mu = 0.05 \) corresponds to the minimization of \( J_n \) (8). Now two paths are well exhibited and identified, and their time variations are followed. Their modulus and phases are given in fig 3d.

V - CONCLUSION

The principle and some results of the estimation of the u.w channel i.r. have been given, particularly for the multipath case, the estimation of delays, amplitudes and phases has been studied. The CCOR step improves SNR and gives a first path separation. The improvement given by the second step (minimization of \( J_n \) and, better, \( J_n \)) enables to solve and characterize close paths. The method performs even when the number of triple parameters is not small (for ex. 6 paths).

This estimation is important in two kinds of objectives:

\( \mu \) the knowledge of the u.w propagation and there is now a large interest in it, for ex. in the international u.w acoustic Tomography Project (6), which is based on the precise time delays estimation.

\( \mu \) the elaboration of efficient detection or communication u.w systems: it is important to know the structure and variations of multipath to compensate them.

This work has been partly supported by French DCN, and also an Ifremer contract.

VI - REFERENCES


Abstract - This paper describes the application of computational fluid dynamics to a hypersonic propulsion system, and serves as an introduction for this session. An overview of the problems associated with a propulsion system of this type is presented, highlighting the special role that CFD plays in the design.

I. INTRODUCTION

CFD has demonstrated some rather significant and spectacular achievements for aeronautical vehicles and their flow fields over the last 15 years. More recently some of these gains have been brought to bear on propulsion systems for aircraft; namely in the complex, wall bounded internal flows with energy addition inside of engines. An extensive activity has been pursued in attempting to validate numerical methods using data obtained from component (inlet, ducts, nozzles, combustors) testing. The computer codes developed have incorporated extensive physical and chemical modeling or closures, as well as utilizing multi-dimensions and sophisticated grid generation and adaptation methods. Due to the extremely complex nature of internal flow of engines, including rotating machinery, the validation and calibration of propulsion codes has proceeded slowly but steadily. With the resurgence of interest in a hypersonic air-breathing aircraft, i.e., the National Aerospace Plane, CFD efforts have been focused strongly on its proposed engine cycle. That cycle as envisioned currently relies on a supersonic combustion ramjet (Mach 6 to 15) used in conjunction with an accelerator up to Mach 6 and a rocket engine from approximately Mach 15 to orbital speed. The challenge to the scientific community is to develop accurate numerical simulations for this type of aircraft and propulsion system. Since scramjets have not been demonstrated on any propulsion system, the cycle must yet be proven feasible. In the absence of any flight data and the meager prospect of obtaining any data above flight Mach numbers of 8 in the near future, CFD becomes the tool of necessity for design of the engine and vehicle. It is worthwhile to point out, that the highly blended propulsion system makes it impossible to consider the engine without considering the influence of the airframe.

II. AIR CAPTURE

It should be noted at the outset that the air reaching the inlet face has experienced a rather trying time from the moment it traverses the shock wave at the aircraft nose. Depending on flight Mach number, the air may be dissociated and ionized as it moves along the underside of the aircraft. This air may undergo catalytic effects at the vehicle wall as well. Chemical and thermal non-equilibrium effects need to be modeled as well as catalicity. At the inlet plane, a substantial boundary layer has been developed on the ramp side of the inlet. It may be laminar or turbulent, or possibly transitional in nature. Shock waves from the cowl leading edge and the inlet sidewalls introduce additional complexities, such as shock-boundary layer interaction, which need to be modeled in the CFD codes. An example of such interactions is shown in a videotape. The computations, which are the result of the work of Benson and Reddy, at NASA Lewis Research Center (LeRC) illustrates a further trial for the captured airflow. The particle tracing shows that the intersection of the cowl and ramp shocks with the sidewall boundary layers causes a movement of the lower energy wall flow towards a narrow region. Cross-sectional inspection of the computed flow field reveals a vortex-like feature. At the throat, this flow behavior extends over a sufficient portion so as to cause concern regarding performance and stability of the inlet. Other rectangular inlets, such as the sidewall compression type, also experience similar effects. In this session we shall hear further discussion on inlets. Additional information is provided by this author in AGARD proceedings. Comparison of the computer inlet flow field and experimental data have shown good general aerodynamic agreement. However, in the regions where strong viscous effects are present, the agreement is marginal. Both transition and turbulence modeling improvements are required. Compressibility effects on turbulence modeling is currently being pursued as well as second moment closure by Shih and his
cohorts at the ICOMP Center for Modeling of Turbulence and Transition at LeRC. Comparisons of heat transfer data on the inlet walls with computed results show significant differences that need to be reconciled if CFD is going to affect scramjet thermal heating design.

III. MIXING AND BURNING

Mixing and combustion of hydrogen in supersonic flow (Mach 1.5 to 7) is the critical issue to be solved for the success of scramjets. A seminal contribution to supersonic mixing was put forward by this author regarding the generation of streamwise vorticity for mixing enhancement. Current generic engine combustors rely on the concept of vorticity generation. The method of generation differs only in detail from that of this author, but not in principle. It employs swept leading edges at angle of attack to the supersonic stream to promote vorticity. Shock vortex interaction was also proposed as a means of mixing enhancement, but it is less influential than vorticity generation. The basic issue revolves about the fact that jet penetration into supersonic flows is limited to about 10 jet diameters; an amount that is insufficient for a combustion chamber. Struts protruding into the stream produce the anticipated and predictable drag, must be cooled, and must be retracted over a portion of the flight range. Some current research centers on the vorticity generation concept mentioned above using swept wall injectors with fuel injection from the back face. CFD development of three-dimensional viscous computer codes with finite rate chemistry are used to compute the mixing and reaction for these devices. Shown in a video is also an unswept configuration for comparison. The computations performed by Moon at LeRC illustrate the extent of the reacting zone for the two configurations. It should be noted that the CFD developed for the combustor does not truly represent the turbulence-chemistry interaction. The chemistry is modeled using a number of chemical steps (12) and a number of species (9). Mean values of temperature and pressure are used to determine the reaction rates. Current research is devoted toward formulating a probability density function model for the chemical reactions. Such a scheme would rely on local instantaneous values of temperature and pressure for the chemical reaction calculations. Again, we shall hear in this session, some further discussion on the mixing and combustion issue.

IV. EXHAUSTING THE AIR

The nozzle, like the inlet, blends into the aerodynamic lines of the vehicle. Here, the underside of the aft portion of the airplane forms a one-sided nozzle surface. On the opposite wall, a short cowl allows the flow to form a free shear layer with the external flow field. Hence, the nozzle dynamics and the shear layer physics and chemistry are radically different than those within our experience. Vehicle speed affects the effective back pressure on the nozzle and causes it to be over- or under-expanded. Shock-shear layer structure is dramatically affected, and can vary from shock impingement on the vehicle to no effect. The composition of the species entering the nozzle and the exit conditions influence the completeness of chemical reaction in the plume. A typical computation by Lai at LeRC using a Reynolds averaged, three-dimensional Navier-Stokes codes is shown in the video. This computation relies on a Baldwin-Lomax turbulence model. One can see the development of sidewall shear layers at the nozzle exit as well as the corresponding features on the cowl surface and shear layer. The nature of the exhaust plume is highly affected by three dimensionality. Only limited data exist for flow field comparison at the present time. There is no doubt, however, that significant closure issues remain to be addressed.

V. CONCLUDING REMARKS

On an overall basis, one can observe that a significant amount of progress has been made on the application of CFD for high-speed airbreathing propulsion systems. Excellent qualitative agreement is the usual picture, with significant discrepancies only in those near wall regions dominated by strong viscous flows. Nonequilibrium air effects and finite rate chemistry are extensively modeled and computed. However, proper turbulence chemistry interaction requires a significant amount of attention. Improvements in turbulence and transition models are also critically needed. I look forward to hearing the presentations in this session on these important issues; I hope you share my enthusiasm.
NUMERICAL SIMULATION OF FLOW THROUGH OPPOSITE AND SIMILAR SWEEP SCRAMJET INLETS

D. J. Singh
Analytical Services & Materials, Inc.
107 Research Drive,
Hampton, VA 23666 U.S.A.

and

Ajay Kumar
NASA Langley Research Center
M.S. 156
Hampton, VA 23665 U.S.A.

Abstract

A comparative numerical study of performance parameters of a similar and an opposite sweep sidewall compression inlet is made. A three-dimensional Navier-Stokes code is used to calculate the flow through these inlets. Results of these calculations are used to compare the two designs for their performance and flow quality. Effects of boundary-layer ingestion on the performance and overall flow features are also investigated.

Introduction

For over two decades, NASA Langley Research Center has been conducting research in developing a viable air-breathing propulsion system for hypersonic flight application. In this flight regime, a supersonic combustion ramjet (scramjet) engine becomes attractive. The inlet of the engine module compresses the flow with the swept, wedge-shaped sidewalls. The sweep of these sidewalls, in combination with the aft placement of the cowl on the underside of the engine, allows for efficient spillage and for good inlet starting characteristics over a range of operating Mach numbers with fixed geometry [1]—[2]. In order to systematically investigate the effects of sweep on the performance of a scramjet inlet, a numerical study is conducted on two equivalent scramjet inlets. These inlets have been designed in such a way that both have the same wetted area [3] but in one inlet design, all of the compression surfaces are swept backward (Fig. 1-a); whereas, in the other design, alternate surfaces are swept backward and forward (Fig. 1-b). A three-dimensional Navier-Stokes code, SCRAMIN [2], is used to analyze the inlet configurations. The code solves the Reynolds-averaged Navier-Stokes equations in conservation form using the MacCormack method.

Results and Discussion

The numerical simulation of the flow through the two inlet configurations described earlier is made for the following freestream conditions

\[ M_\infty = 4.5, \quad T_\infty = 200^\circ K, \quad p_\infty = 3376.86 \quad N/M^2 \]

The calculations were made for uniform flow entering the inlet configurations and with a 10% (of inlet height) and a 20% thick entering boundary layer to determine the effect of boundary-layer ingestion on the performance of the similar and opposite sweep inlet. Due to space limitation, only representative results are shown here; the detailed results are available in Ref. [3].

Figure 2 shows the pressure plots in three longitudinal planes for the two configurations with uniform flow entering the inlet. It shows the shock and expansion waves and their interactions. This figure shows an advantage of the opposite sweep over the similar sweep. The sidewall shocks in the opposite sweep inlet do not intersect with the sidewall boundary layer in a given swept, constant-area cross-section. Therefore, any blockage created by the shock-induced separation of the boundary layer is not as damaging to the inlet performance as in the case of similar sweep inlet where the shock/boundary layer interaction and associated separation takes place in a swept, constant-area cross-section.

Figure 3 shows plots of Mach number, stagnation pressure, and static pressure in a cross plane near the throat for the two configurations with 20% thick boundary layer entering the inlets. It shows approximately half of the cross-section is now filled with the viscous, nonuniform flow because the 20% thick entering boundary layer near the top wall is being squeezed into the throat region which is four times smaller in width than the entering cross-section.

Using the flow field results, calculations were also made for inlet performance quantities such as the average throat Mach number, total pressure recovery, axial thrust, and mass capture. The calculations showed that there was little difference in the average throat Mach number and total pressure recovery of the two inlets. However, the mass capture significantly increased for the opposite sweep inlet. Thus, the detailed flow field results suggest that the overall impact of the opposite sweep is quite favorable on the inlet performance.

References

Figure 1: Generic inlet configuration for the study of sweep effects.

Figure 2: Static pressure contours at three height locations with uniform entering flow.

Figure 3. Plots of inlet performance parameters near throat with 20% thick entering boundary layer.
NUMERICAL STUDY ON SUPERSONIC CHEMICALLY REACTING FLOWS
SATORU OGAWA, YASUHIRO WADA AND TOMIKO ISHIGURO
Computational Sciences Division, National Aerospace Laboratory, Chofu, Tokyo, Japan

Abstract This paper describes the numerical techniques to solve the supersonic chemically reacting flows. The higher-order upwind scheme based on a generalized approximate Roe's Riemann solver is used, and a fully implicit time integration method is used to accelerate convergence rates. As the numerical examples, the hypersonic flow around space vehicle, and the supersonic combustion in SCRAM jet engine are presented.

I. INTRODUCTION

The basic research and development is driven forward in the NAL for the space plane which are capable to go into the space and return with ease. The most important subjects in this development would be, the precise evaluation of aerodynamic and aerothermodynamic characteristics in hypersonic region where the real gas effect is dominant, and the development of the supersonic combustion RAM (SCRAM) jet engine. No ground-based experimental facilities can fully duplicate the conditions that these vehicles will encounter in the upper atmosphere, hence the numerical simulation is expected to be one of the most promising method in the study of hypersonic chemically reacting flows. In this study, chemical nonequilibrium flows are solved by the use of higher-order upwind scheme. This scheme[1] is based on a generalized Roe’s approximated Riemann solver, and arbitrary nonequilibrium effects are treated in a unified formulation. A fully implicit time integration method is used to accelerate convergence. As numerical examples, chemically reacting hypersonic flows around the Space Shuttle, and the supersonic combustion in the SCRAM jet engine are solved.

II. BASIC EQUATIONS

The basic equations of compressible chemically reacting flow are written in the weak conservation form:

$$\frac{\partial q_i}{\partial t} + \frac{\partial F_i(q)}{\partial x^k} = s_i,$$

where $q$ is a conservative quantities per unit mass, and $s$ is its corresponding source term.

$$q = [\rho, \rho u, \rho v, \rho w, E, \rho Y_1, \rho Y_2, \ldots, \rho Y_n]^T,$$

where $E$ is the total energy, $E = e + 1/2 \rho V^2$, and the internal energy $e$ is given by:

$$e = \rho \sum_i Y_i (i \sum_i \rho Y_i) + 1/2 \rho V^2.$$

In this study, chemical nonequilibrium flows are solved by the use of higher-order upwind scheme. This scheme[1] is based on a generalized Roe’s approximated Riemann solver, and arbitrary nonequilibrium effects are treated in a unified formulation. A fully implicit time integration method is used to accelerate convergence. As numerical examples, chemically reacting hypersonic flows around the Space Shuttle, and the supersonic combustion in the SCRAM jet engine are solved.

III. NUMERICAL SCHEME

In recent years, upstream difference schemes have yielded a great success in flow computation. Most of these schemes make use of the exact or approximated solution of the Riemann problem as a building block. Among them, Roe’s approximated Riemann solver is one of the most promising method, and in this paper, a generalized Roe’s approximate Riemann solver for nonequilibrium flows is used. Chakravarthy Osher postprocessing TVD scheme is used to make the higher-order scheme. It is more efficient than MUSCL scheme for a nonequilibrium flow problem, because the latter method needs Newton iterations at each cell interface to calculate the value of temperature from conservative variables in the case of chemically reacting flows.

Generally nonequilibrium flow is very stiff problem, so that it is desirable to treat every term implicitly. In this paper each convective block operator is diagonalized so that the block matrix operation is reduced to the scalar one. Further, in order to enhance robustness, near by shock waves only chemical source terms are implicitly treated and the rest explicitly. This patched method[2] is easily constructed by replacing tri-diagonal scalar operators by a unit matrix.

IV. EXAMPLES OF NUMERICAL COMPUTATIONS

A. Hypersonic flow of real gas

Since the space plane flies more than ten times as fast as the sonic speed, the extremely strong shock wave appears ahead. The strong compression behind the strong shock wave causes the high temperature more than ten thousand degree near the plane. Thus the nitrogen and oxygen in the atmosphere dissociate, consequently the usual assumption of perfect gas no longer holds good and it becomes necessary to include the effect for the real gas. In this example the elemental reactions
for 7 components of \( N, O, N, O, NO, NO', e \) are considered as the dissociation in order to include this real gas effect. In the case of the space plane, when the shock wave strikes the wing, the plane suffers the very severe aerodynamic heating. Therefore the prediction for the situation of shock wave is very important and the correct evaluation for the real gas effect is necessary. In Fig.1 are shown the numerical solution for the hypersonic flow of Mach number 15.7 around the Space Shuttle with the real gas effect included. The distribution for the mole fraction of the atomic oxygen produced by the dissociation is displayed in the figure. The atomic oxygen near the nose of the plane is transported with fluid to gather to the center parts on both the upper and lower surfaces separately. Further on the upper surface it is transported with the separated fluid to spread over the plane again.

B. Chemically-reacting flow in a combustor

As the computation of chemically reacting flow has been possible with the progress of computers, the computational condition of this example is almost adapted to the actual experimental one for the SCRAM jet engine. The foundation of physical phenomenon in the SCRAM engine is that the hydrogen blows up into the high temperature gas and burns. For the combustion a new chemical reacting model in the Westbrook reacting model 9 chemical species of \( N, N_2, O_2, OH, H_2O, H, O, H_2O, \) and \( HO \) are considered and 17 elementary reaction steps are contained. The chemically reacting flows in the combustor where the hydrogen blows up have been numerically solved using the TVD scheme for the governing equation system above stated. An example is shown in Fig.2, where iso-Mach contours are shown. Thus by the CFD it is possible to see the flow details such as the Mach disk formed by the blow, which is difficult to measure by experiments. Since the problem of turbulence in reacting flows remains almost unsolved, however, the reacting ratio does not show a good agreement between the numerical solution and the experiments. The reason is that the reacting ratio greatly depends on the mixing of hydrogen and oxygen, which the turbulent diffusion governs. The problem of turbulence in reacting flows would be an important researching theme in the future.

V. CONCLUDING REMARKS

By the development of CFD till now, it has been possible to obtain the numerical solutions which hold good to some extent for a large variety of problems. The problems for not only chemical reaction but also radiation and electromagnetic fluid dynamics can be numerically solved without difficulty if costing much time. It may fairly be said that the final problem still remained is the turbulence that is a remarkable characteristics of the non-linear fluid motions.

References
RAPID PHASE CHANGES IN ISO-OCTANE IN THE GENERAL VICINITY OF THE CRITICAL POINT

S. C. Gulen, H. J. Cho, A. Hirsa, P. A. Thompson, M. Moran

Department of Mechanical & Aerospace Engineering
Rensselaer Polytechnic Institute
Troy, NY, U. S. A.

Abstract
Stationary states at high temperatures and pressures in Iso-Octane (2,2,4-Trimethylpentane) were achieved behind a compression shock wave reflected from the shock-tube end wall. The end states cover a broad region including vapor, liquid, and mixture phases. Measurements of shock velocity, pressure, and temperature were performed together with extensive photographic observation of the final states through a sapphire window mounted at the end of the shock-tube observation chamber. In general, phase changes resulting from the shock compression of a retrograde substance are predicted reasonably well with the equilibrium Rankine-Hugoniot model. Photographic observations of the end states show a rich variety of two-phase vortex rings, depending on initial conditions and shock strength.

Theory & Results
The x-t diagram of the reflected shock system is shown in Figure 1. Figure 2 shows a detailed reflected shock adiabat on a P-v surface. The adiabat crosses the phase boundary but the process is so fast that the condensation is delayed until the limit of supersaturation, Wilson line, is reached whereupon a spontaneous collapse of the metastable state occurs. If the shock strength is such that the end state lies below point 2 (triple point) two distinct discontinuities are observed: a forerunner, “dry” shock and a condensation discontinuity [3]. In our experiments the shock Mach numbers are high enough that the end states lie above the triple point and a single liquefaction shock front exists.

Introduction
Adiabatic, pressure driven, finite amplitude waves resulting in phase changes - from vapor to liquid or liquid to vapor - have been observed in retrograde fluids. Unlike thermally driven phase transitions in regular fluids such as water, these waves result from a jump in pressure. Recent work on the subject includes complete and partial liquefaction shocks [1], rarefaction shock from a critical state [2], shock splitting [3], and mixture evaporation rarefaction shock [4]. In this work rapid phase change phenomena in the general vicinity of the critical point are investigated. Thermodynamic, vapor-liquid critical point as described by modern power laws [5] is a singular point of infinite isothermal compressibility and constant volume specific heat (hence, in principle, zero soundspeed.) A recent study of nonequilibrium, near-critical states in shock tube experiments revealed a minimum in the soundspeed (about 10 m/s) and disappearance of two distinct phases at a pressure ca. 25% above the critical value [6].

Figure 1. x-t diagram of the shock-tube flow. The closed end of the test section is at right. Arabic numerals designate test fluid states. I=incident shock, R=reflected shock, CS=Contact surface, D=Driver gas initial state. Not to scale.

Figure 2. Pressure-volume diagram of the reflected shock. EA=equilibrium adiabat (liquefaction shock); DA=dry adiabat (nonequilibrium, metastable, supersaturated vapor); R=Rayleigh line; σ=saturated vapor boundary; W=Wilson line (line of critical supersaturation); 2=triple point (see [3] for details.)

In the calculations Rankine-Hugoniot equation is used together with a virial-type, corresponding states equation of state which is modified to represent the near-critical region more accurately. Figure 3 shows experimental data for a reflected shock system which passes through the theoretical critical point. The agreement with the calculations is quite good. Nonetheless, there is a systematic deviation in temperatures measured by very thin (5×10⁻⁴ in diameter), fast response thermocouples. They are lower than calculated values by ca. 5 to 10°C depending on the shock strength.
In addition to the measurement of temperature and pressure, photographic observations are made for different incident shock Mach numbers. Four different initial conditions are chosen with resulting reflected shock adiabats shown in Figure 4. Particular
growth take place. The compression rate across the frozen part of the shock is so high that the nucleation process lags. There are mainly three relaxation processes, inertial relaxation (there is a velocity slip between droplets and suddenly decelerated vapor), droplet temperature relaxation, and vapor thermal relaxation. To give a

![Reflected shock adiabat](image)

Figure 3. Pressure and temperature behind reflected shock. \( T_0=130 \) °C, \( P_0=0.445 \) bars. \( \sigma = \) vapor pressure line, \( \sigma \text{p.} = \) critical point.

emphasis was given to the end states where the shock adiabat crosses saturated vapor and liquid boundaries. Designation of these regions are as shown in Figure 4 and a description of the nature of the associated phase change phenomena is given in Table 1. Except from the region on the saturated vapor boundary far from the critical point (Figure 5 a), two-phase vortex rings are consistently observed both on saturated vapor and liquid boundaries. The vortex rings are thought to be a direct result of steep pressure and density gradients across the shock wave acting upon the nucleation clusters. In regions SL1 or SL2 (Figure 5 b&c, respectively), a group of large, turbulent vortex rings amongst thousands of tiny, small-scale, newly born vortex rings have been observed. Vortex rings which are formed become turbulent within a few tens of microseconds and turbulent vortex rings grow linearly with time (Figure 6.) Vortical structures near the critical point (Figure 5 d&e) are even more turbulent and diffuse. It is important to note that the critical point we are referring to is the one calculated by the equation of state. The proximity of the observed states to the actual, non-equilibrium critical point is open to discussion.

Unique and fascinating structures are observed in region SV3 (Figure 5 f.) These are circular, two-phase structures with apparent rays radially emanating from the center. As the end states approach the critical point, i.e. regions SV4 and SV5 (Figure 5 g&h, respectively), these ray structures still can be seen.

The structure of the liquefaction shock front can be divided into two parts. A frozen discontinuity dominated by the viscosity and thermal conductivity, typically a few mean free paths wide, followed by a considerably longer relaxation zone where nucleation and droplet

<table>
<thead>
<tr>
<th>Shock Adiabat</th>
<th>States behind the reflected shock</th>
<th>Description of phase change</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Liquid Saturation Boundary : SL1</td>
<td>Well defined two-phase vortex rings in liquid or dense gas Condensation in a normal fashion in vapor</td>
</tr>
<tr>
<td>B</td>
<td>Liquid Saturation Boundary : SL1</td>
<td>Well defined two-phase vortex rings in liquid or dense gas Condensation in a normal fashion in vapor</td>
</tr>
<tr>
<td>C</td>
<td>Near Critical Region : NC Vapor Saturation Boundary : SV3</td>
<td>Featherly, more diffuse and turbulent two-phase vortex rings Interesting ray structure in condensation in vapor</td>
</tr>
<tr>
<td>D</td>
<td>Vapor Saturation Boundary : SV3</td>
<td>Ray structure, two-phase vortex rings in vapor Ray structure, two-phase vortex rings in vapor</td>
</tr>
</tbody>
</table>

Table 1. Summary of phase changes on the vapor and liquid-saturation boundaries.

A numerical example, an incident shock \( (M_p=2.46) \) produces a two-phase mixture \( q = 0.33 \) after reflection from the end wall. The width of the frozen discontinuity is computed to be approximately \( 0.05 \) \( \mu m \) (ca. 5 mean free paths) with a resulting supersaturation of \( S=2.3 \).

Approximate calculations showed droplet temperature relaxation (ca. \( 4x10^{-15} \) sec.) to be much faster than inertial relaxation which is itself approximately one order of magnitude faster than vapor thermal relaxation \( (7x10^{-10} \text{ and } 1x10^{-9}\text{ sec.}, \text{respectively}) \). Numerical integration of the equations of motion coupled with a suitable nucleation and droplet growth model will yield more accurate description of the liquefaction shock structure.

![Shock Adiabat Diagram](image)

Figure 4. Four reflected shock adiabats A, B, C, and D along which experiments are done with increasing shock Mach numbers. SL1, SL2, SV1, SV2, SV3, SV4, SV5, and NC designate regions where extensive photographic observations are made (see also Table 1.) Drawing is not in scale.
Conclusion
Photographic observations show a variety of new and rich phenomena associated with rapid phase changes across the liquefaction shock front. Two-phase vortex rings are formed by the pressure gradient across the shock front acting upon the nucleation sites which are denser than the surrounding gas. Vortex rings quickly become turbulent upon formation and grow linearly thereafter. Measured downstream pressures agree well with equilibrium calculations, whereas temperature measurements are lower than computed values. The results are mainly qualitative and current efforts focus on quantitative description of the vortex formation and liquefaction shock structure.

Figure 6. Average vortex ring diameter versus time. Vortex rings are produced in Region NC. $T_0=130^\circ$C, $P_0=0.445$ bars.

References

Figure 5. Photographs taken during the rapid phase change experiments through a sapphire window mounted at the end wall of the shock-tube. See Figure 5 and Table 1 for region designations. a) Region SV1, $M_0=2.46$; b) Region SL1, $M_0=2.94$; c) Region SL2, $M_0=2.85$; d) Region NC, $M_0=2.82$; e) Region NC, $M_0=2.87$; f) Region SV3, $M_0=2.30$; g) Region SV4, $M_0=2.33$; h) Region SV5, $M_0=2.78$. 

569
Abstract — We examine steady transonic flows of Bethe–Zel'dovich–Thompson (BZT) fluids. An extension of the transonic small disturbance equation, valid in the neighborhood of one of the zeros of the fundamental derivative, is presented. Numerical solutions reveal that the natural dynamics of these fluids may result in a significant increase in the critical Mach number. We also report transonic flows involving both expansion and compression shocks in the same flowfield.

I. INTRODUCTION

Transonic flows are inherently nonlinear and are typically accompanied by shock formation in the hyperbolic portions of the flow. Because of the strong adverse pressure gradients associated with compression shocks, shock-induced separation is a major concern in the design of transonic turbomachinery. Such separation is a major loss and vibration mechanism. The process of shock formation is due to the intrinsic nonlinearity of the fluid. As a result, it has been assumed that all flows behave generally the same as predicted by the perfect gas theory and most work has addressed blade design to minimize these effects.

The appropriate measure of the intrinsic nonlinearity of the fluid is the thermodynamic parameter

\[ \Gamma = \frac{V^3}{2s^2} \left( \frac{\partial^2 p}{\partial V^2} \right)_s \tag{1} \]

where \( V, p \) and \( s \) are the fluid specific volume, pressure and entropy. The quantity

\[ a = \sqrt{V \left( \frac{\partial p}{\partial V} \right)_s} \] \tag{2}

is the thermodynamic sound speed. Here we follow Thompson [1] in referring to (1) as the fundamental derivative of gasdynamics. Inspection of (1) indicates that \( \Gamma \) is a measure of the curvature of the isentropes.

Recent studies have shown that the gasdynamics of fluids having relatively large specific heats may be qualitatively different than that of lighter substances such as air and water. The main objective of the present study is to examine the behavior of these fluids in the transonic regime. In particular, it will be shown that use of these fluids results in significant increases in the critical Mach number, thereby decreasing the range of flow speeds at which the undesirable transonic flow effects are observed.

The fluids of interest here are those for which \( \Gamma < 0 \) for a finite range of pressures and temperatures. The conditions under which \( \Gamma < 0 \) where first given by H. A. Bethe [2] and Ya. B. Zel'dovich [3], who demonstrated that fluids having relatively large specific heats will have a region of negative \( \Gamma \) in the fluid's dense gas regime. The general region where \( \Gamma < 0 \) is depicted in Figure I where the isentropes of normal decane have been computed and plotted on a \( p-V \) diagram. The regions of downward curvature correspond to \( \Gamma < 0 \). Further examples of commonly encountered fluids having \( \Gamma < 0 \) have been provided by Thompson and co-workers [5], [6], and Cramer [7]. Because of the contribution of the early workers in this area, we refer to fluids possessing a region of \( \Gamma < 0 \) in the single-phase regime as Bethe–Zel'dovich–Thompson (BZT) fluids.

The significance of BZT fluids is that the well-known compression shocks of the perfect gas theory violate the entropy inequality and disintegrate into centered fans if \( \Gamma < 0 \). Expansion shocks, normally forbidden in the perfect gas theory, not only form from smooth waves but are seen to satisfy the entropy inequality in flows having \( \Gamma < 0 \) everywhere. It therefore appears that the natural dynamics of BZT fluids may lead to a reduction or even elimination of shock–induced separation.

Figure 1. Computed isentropes for n-decane. Gas model is the HBMS [4] equation of state.

Thus, there is likely to be advantages in the use of BZT fluids even in supercritical flows. For further details of the remarkable dynamics of BZT fluids, we refer the reader to the surveys found in References [1], [8]–[9].

II. THE TRANSONIC SMALL DISTURBANCE EQUATION

We consider small two-dimensional disturbances to a near-sonic flow of a BZT. The usual assumptions of the transonic small disturbance theory will be supplemented by the condition that the thermodynamic state of the freestream is in the vicinity of one of the zeros of the fundamental derivative. Such zeros form the boundary between the positive and negative \( \Gamma \) regions and are recognized as the inflection points of the isentropes in Figure I. With this freestream state the small perturbations caused the thin blade can take the flow from regions of positive to negative \( \Gamma \) and vice versa. In this sense, the flow will be qualitatively similar to those involving larger amplitudes. Furthermore, the most complex and interesting features occur when the nonlinearity is mixed. When these assumptions are incorporated, Cramer [8] has shown that the extension of the classical transonic small disturbance equation is

\[ \left[ \frac{M^2}{\rho} + 1 + 2^2 \frac{\hat{\phi}_x}{\hat{\phi}_x} - \Lambda \frac{\phi_1^2}{\phi_1} \right] \phi_{xx} = \phi_{yy} \] \tag{3}

with the usual boundary conditions at the blade and at infinity. Here \( \phi \) is a nondimensional velocity potential, \( \lambda \) is the scaled distance normal to the freestream.
The Mach number and fundamental derivative in the freestream are denoted by \( M_\infty \) and \( \Gamma_\infty \), respectively. The parameter

\[
\Lambda \equiv \rho \frac{\partial \Gamma}{\partial p} \left|_{a^*}\right.
\]

(4)

where \( \rho \equiv V^{-1} \) is the fluid density, is a second nonlinearity parameter. This is usually evaluated at the state corresponding to the zero in the local value of \( \Gamma \), without loss of accuracy to the overall approximation. Near the large density zero this parameter is negative near the low density zero of \( \Gamma \).

III. RESULTS

Numerical solutions to (3) have been generated through use of an extension of the Murman–Cole scheme. The boundary conditions were those of a circular airfoil. The results of these calculations are depicted in Figures 2–4. In order to check the scheme, calculations for a classical (\( \Lambda=0 \)) case were carried out. Here \( \Gamma_\infty = 0.4, M_\infty = 0.9 \) and the Mach numbers indicates that the flow remains entirely subsonic. That is, the use of a BZT fluid in the neighborhood of one of its zeros have driven the critical Mach number to values above 0.9. Physically, the increase is due to the fact that the expansions over the top of the blade has shifted the flow into the \( \Gamma < 0 \) regime before the sonic state is reached. As pointed out in References [1], [6], and [10] acceleration through the sonic condition is then difficult, if not impossible. In order to save space, we refer the reader to these previous studies for a detailed account.

The increase in the critical Mach number is by no means an isolated case. Further detailed studies suggest that blade configurations giving rise to a critical Mach number of 0.69 in air may correspond to a critical Mach number over 0.98 in many of the BZT fluids described in Reference [6] or [7].

As a final example, we have retained the same upstream thermodynamic state as used in Figures 2–3, but have raised the Mach number to 0.93, approximately. In this case, the flow is able to achieve sonic conditions before the \( \Gamma = 0 \) point is reached. The result is plotted in Figure 4. In this supercritical flow, two shocks appear. The first (leftmost) is an expansion shock which is followed by a compression shock. This is in marked contrast to classical theory where no more than one shock can occur.

The preceding has introduced a modified form of the transonic small disturbance equation valid when the freestream is in the vicinity on one of the zeros of the fundamental derivative of a BZT fluid. The numerical results demonstrate that the natural dynamics of BZT fluids can give rise to significant increases in the critical Mach number as well as qualitative differences in the details of supercritical flows.

The author would like to thank G. M. Tarkenton for providing Figures 2 and 3. \( \Gamma_\infty = 0.4, \Lambda = 1.0 \) and \( M_\infty = 0.93 \).

IV. SUMMARY

REFERENCES

A NEW HIGH-RESOLUTION GLOBAL SPECTRAL MODEL FOR MEDIUM-RANGE NUMERICAL WEATHER PREDICTION

CLIVE TEMPERTON
European Centre for Medium-Range Weather Forecasts
Shinfield Park, Reading, Berkshire RG2 9AX, U.K.

Abstract. A new high resolution spectral model has been developed for the production of medium range forecasts at ECMWF. Considerable efficiency gains were required to make this model operationally practicable; these were realized principally by the introduction of a reduced computational grid and a semi-Lagrangian integration scheme.

Current plans for the operational ECMWF spectral model (Simmons et al., 1989) include a doubling of the horizontal resolution from T106 (a triangular truncation at wavenumber 106) to T213, with a corresponding increase in the number of vertical levels from 19 to 31.

Two new ingredients of the numerical integration procedure are essential in order to produce timely operational forecasts with this high-resolution model, given the constraints of the present computer system.

The first is the use of a reduced Gaussian grid (Hortal and Simmons, 1991) for the computation of nonlinear terms; in this grid, the number of points per latitude row is decreased towards the poles so that the east-west gridlength remains approximately constant. The computation per time step is thereby reduced by around 25%, with very little impact on the resulting forecast.

The second ingredient is the introduction of a semi-Lagrangian semi-implicit time-integration scheme, which overcomes the stability criterion of the conventional Eulerian treatment of advection (spectral in the horizontal, finite difference in the vertical). The semi-Lagrangian formulation is basically similar to the fully three-dimensional interpolating version of Ritchie (1991), there are however additional complications resulting from the use of a hybrid vertical coordinate rather than the sigma-coordinate of Ritchie's model. On the other hand, the vertical discretization by fine differences and the purely algebraic elimination between variables in the solution of the semi-implicit equations both lead to some 500hPa 2CJG Day 1 T213.31

Tests have shown that the Eulerian version of the T213 31-level model requires a 3-minute timestep to maintain stability, while the semi-Lagrangian version remains stable and accurate with a 20-minute timestep. In addition to the gains already obtained from the use of the reduced grid, the semi-Lagrangian scheme yields a further factor of about 5 in the efficiency of the model. To demonstrate that the semi-Lagrangian version of the T213 31-level model gives essentially the same forecasts as its Eulerian counterpart, Figs. 1 and 2 show the corresponding 3-day forecasts of the 500 hPa height field starting from the operational ECMWF analysis at 12Z on 15th April 1990.

REFERENCES


500hPa 2CJG Day 3 T213.31
INTEGRATION OF A GLOBAL MULTILEVEL MODEL USING A VECTOR SEMI-LAGRANGIAN SCHEME WITH A MULTIGRID SOLVER

J. R. Bates, S. Moorthix and R. W. Higgins
Global Modeling and Simulation Branch
Code 911
NASA/Goddard Space Flight Center
Greenbelt, MD 20771

Abstract

A three-dimensional semi-Lagrangian semi-implicit two-time-level finite-difference integration scheme for the primitive equations of atmospheric motion on the sphere is presented. A trajectory-centered discretization of the governing equations is used, the momentum equation being discretized in vector form before being resolved into components. For the horizontal differencing a C-grid is used, while a Lorenz-grid in \( \sigma \)-coordinates is used in the vertical. The discretised equations, which involve a coupling between all levels, are decoupled by means of a linear transformation, resulting in a set of \( K \) (\( K \) number of levels) two-dimensional elliptic equations to be solved. With an appropriate formulation of the discretized governing equations, these elliptic equations have a form identical to that encountered in an earlier shallow water model (Bates et al., 1990). The two-dimensional multi-grid solver used in the shallow water case can thus be used to provide an efficient solution for the multilevel case.

A linear stability analysis of the scheme on an \( f \)-plane in the absence of a mean flow shows that the scheme is unconditionally stable.

Numerical integrations are performed using an adiabatic version of the model, both with and without orography and divergence damping. The results of these integrations with varying time steps and with both idealized and observed initial conditions will be presented.

Reference


xUniversities Space Research Association
A CLASS OF MONOTONE INTERPOLATION SCHEMES

PIOTR K. SMOLARKIEWICZ and GEORG A. GRELL

National Center for Atmospheric Research

Among a variety of available advection schemes, the dissipative (forward-in-time) algorithms are the most suitable for practical applications. Since the effective velocity field is constant for every point of interest to the interpolation procedure, these advection schemes retain their formal accuracy of the constant coefficients limit. Constancy of the effective velocity in an arbitrary-dimensional problem allows for a straightforward alternate-direction implementation of one-dimensional advection schemes without introducing errors characteristic of the time-split advection procedures in variable flows.

Insofar as the linear dissipative advection schemes are concerned, there is no particular gain from such an exercise as the resulting interpolators may be alternatively derived with the help of more traditional arguments invoking either the truncated Taylor formula or Lagrangian polynomial fitting. However, when the preservation of monotonicity and/or sign of the interpolated variable is essential, then the approach adopted becomes useful. For instance, a variety of monotone (and sign-preserving) interpolation schemes of different overall accuracy and complexity levels may be generated using Flux-Corrected-Transport (FCT) versions of high-order-accurate dissipative schemes. The utility of such monotone interpolators is illustrated with examples of applications to selected problems of atmospheric fluid dynamics.

ABSTRACT

The practice of computational fluid dynamics often requires accurate as well as nonoscillatory interpolation procedures. Such procedures, often referred to as shape-preserving interpolation, may be employed to design monotone advection transport algorithms. This paper poses an inverse problem. A variety of monotone advection schemes with attractive properties has been developed over the last two decades abstracting from any arguments that invoke explicit interpolation procedure. Our goal is to provide a formalism allowing the exploitation of these advection algorithms as shape preserving interpolators. The central theoretical issue concerns a formal equivalence of the advection and interpolation operators on discrete meshes. Through elementary arguments exploiting either the Stokes theorem or the untruncated Taylor formula at 0-th order of expansion, one may show that the solution to an interpolation problem can be expressed as a formal integral of the advection equation. As a consequence, the interpolating operator on a discrete mesh may be represented by an advection scheme, in which the local Courant number vector is replaced by the normalized displacement between a grid point and a point of interest to the interpolation procedure. The accuracy of the resulting interpolation scheme is that of the advection scheme employed.
Terrain-following vs. a blocking system for the representation of mountains in atmospheric models

Fedor Mesinger and Thomas L. Black
UCAR Visitor Research Program
National Meteorological Center
Washington, DC 20233, U.S.A.

Abstract. Issues of the numerical representation of mountains in atmospheric models are reviewed. An example is given illustrating the origin of the most frequently considered type of errors resulting from terrain-following coordinates. Finally, a forecast obtained with the model using a so-called step-mountain coordinate is compared with that obtained when the same model is run with a standard terrain-following (sigma) coordinate.

I. INTRODUCTION

Since its introduction by Phillips in late fifties the terrain-following vertical coordinate has been by far the most predominant in atmospheric modeling. While a number of its problems were recognized relatively early, the simplicity it offered for the representation of mountains seemed to be more than a sufficient compensation for the problems. Furthermore, for longer than a decade, various methods were being devised to minimize the errors. These techniques were addressing the finite-difference methods almost exclusively used at that time.

In the early eighties, however, it became increasingly doubtful that the errors associated with terrain-following coordinates could indeed be adequately minimized. It was shown that the accuracy of the calculation of the pressure gradient force over steep mountain slopes is likely to deteriorate rather than improve with an increase in vertical resolution. On the other hand, a different technique using step-like representation of mountains with approximately horizontal coordinate surfaces was proposed and appeared to be an attractive alternative. For a comprehensive review of these various techniques proposed up to the mid-eighties as well as a description of the step-mountain system the reader is referred to Mesinger and Janjic (1985).

Efforts aimed at reducing or eliminating the pressure gradient force errors continued (e.g., Zheng and Liu 1986, Carroll et al. 1987). At the same time, new examples of potentially large errors were reported with regard to the terrain-following coordinate scheme in spite of a sophisticated design (Mesinger and Janjic 1987). A comprehensive step-mountain ("eta") coordinate model was developed and in extensive tests has been shown to be competitive with an operational state-of-the-art terrain following coordinate model. While evidence was presented indicating that the numerics of the model were primarily responsible for the improved results (Mesinger et al. 1990), no assessment was made of the extent to which the improvement came from the vertical coordinate as opposed to other numerical features of the model.

A recent study by Janjic (1989) shows that in spectral models, now dominant in global weather and climate simulations, pressure gradient force errors might be still greater than in finite-difference models. Only the terrain-following formulation appears to be practical for spectral models at present however.

II. AN ASYMPTOTIC LIMIT OF THE ERROR

An example which lends itself to analytic treatment and is useful for assessing errors of various schemes is that of a resting hydrostatic atmosphere in which the pressure gradient force must be zero. Given a numerical scheme, an assumed temperature profile, and the mountain slope, the pressure gradient force can typically be calculated in a straightforward way which hopefully indicates the magnitude of the error.

A recent calculation of this type for three schemes and three temperature profiles has been reported by Mesinger and Janjic (1987). Of the three schemes, the latest is the "\( \Phi \)-conserving" scheme of Arakawa and Suarez (1983) expanded to include horizontal differencing. Rather large errors were obtained, at some of the levels and temperature profiles, for all three of the schemes. In particular, a convergence problem was identified in all three of the schemes in the sense that no tendency was visible for the general magnitude of the error to decrease with increasing vertical resolution.

The three schemes considered are identical for the case of an isentropic atmosphere where the potential temperature \( \Theta = \Theta + \text{const} \). Considering the case of zero pressure at the top of the model atmosphere \( (P_T = 0) \) the asymptotic value of the error as thicknesses of the layers \( \Delta \sigma \) tend to zero is

\[
- \frac{1}{\Delta x} \frac{\partial \Phi}{\partial \sigma} \left| \frac{R \Theta}{P_0} \left( \frac{\Delta x}{P_0^k} \right)^{\frac{1}{\kappa}} \left( \frac{P_0^k}{P_0^x} \right) \frac{\Delta x}{P_0^x} \right| \sigma. \quad (1)
\]

Here \( \Delta x \) and \( \Delta \sigma \) are the centered two-point difference and averaging operators, respectively, applied along the direction of the \( \sigma \) axis; \( \Theta \) is geopotential; \( \kappa \) is the vertical index, increasing downward, identifying quantities defined for the sigma layers; \( R \) is the gas constant; \( P_0 \) is the reference pressure used to define the potential temperature; \( \kappa \) is \( R/C_p \), where \( C_p \) is the specific heat at constant pressure, and \( P_0^x \) is the surface pressure. Furthermore, denoting the partial derivative and its centered two-point difference analog by \( \partial \) and \( \partial \), respectively, note that in the limit as \( \Delta x \) also approaches zero

\[
\frac{1}{\kappa} \frac{\partial P_0^x}{\partial \sigma} \rightarrow \frac{\Delta x}{P_0^x} \frac{\partial P_0^x}{\partial \sigma} = \frac{\Delta x}{P_0^x} \frac{\partial P_0^x}{\partial \sigma}.
\]

575
Thus, in that limit, there will be no pressure gradient force error. However, for finite $\Delta x$, the right hand side of (1) will be different from zero, in spite of the exact finite-difference hydrostatic equation of the three schemes in the considered isentropic atmosphere case.

Attempting to reduce the problem by increasing the horizontal resolution actually tends to worsen the situation since the accompanying incorporation of more realistic mountains simply introduces more sloping terrain. Thus, representation of mountains seems to be an issue of a steadily increasing priority to atmospheric modelers.

III. A FORECAST EXAMPLE

It is not obvious to what extent pressure gradient force errors of a specific idealized example are relevant in actual atmospheric simulations. In addition, other problems are associated with terrain following coordinates. A potentially serious problem is that of horizontal advection with sloping coordinate surfaces such that the vertical velocity relative to these surfaces is required to compensate for their slope.

A study of model performance in actual weather situations thus seems to be the ultimate answer. It is our intention to complete such a study, an example will be shown here. A comprehensive (eta) prediction model was employed which can be used either with a terrain-following or with the step-mountain coordinate, the code and the schemes being the same. The case to be shown was the first case we looked at from the point of view of sensitivity to the vertical coordinate after an extended period of model development. It involves a major cold air outbreak along the eastern slopes of the Rockies.

The U.S. National Meteorological Center surface analysis for 1200 UTC 2 February 1989 is shown in the upper panel of Fig. 1. The 36-h sea level pressure forecast valid at the same time and obtained with the model using the terrain-following ("sigma") formulation is shown in the middle panel. Finally, the 36-h forecast valid at the same time obtained using the step-mountain (eta) coordinate is shown in the lower panel.

One feature favoring the eta result is the reduced noisiness of the eta integration with 21 centers printed as compared to 27 centers of the sigma map. Note that these maps are printed without the usual smoothing prior to output so that unrealistic centers are obtained as the result of model noise. Another feature is the more accurate simulation of the southward extent of the cold air east of Rockies. For example, note that the sea level pressure over North Dakota in the eta integration is about 4 mb higher than it is in the sigma integration. Still higher sea level pressures are seen in the analyzed map. We intend to analyze such differences further and to report on our results more extensively at a later occasion.

REFERENCES

Boundary-Layer Meteor., 41, 149-169.


576
NUMERICAL METHODS FOR A UNIFIED FORECAST/CLIMATE MODEL

H.J.P. CULLEN
Meteorological Office, London Road, BRACKNELL, Berks. RG12 2SZ, U.K.

Abstract. Finite difference methods which combine the efficiency required for forecast models with the conservation properties required for long-term climate integrations are described. Methods of using non-oscillatory advection schemes within this framework are discussed.

I. INTRODUCTION

Numerical methods for forecast models are usually selected on the basis of efficiency and accuracy. Finite difference methods within this category are the semi-implicit and split-explicit methods, with further improvements available from using semi-Lagrangian advection. In climate change experiments global conservation properties must be enforced, and measures taken to ensure satisfactory long-term behaviour. Correct treatment of the energetics is one necessary requirement.

This paper extends the split-explicit scheme of Cadd (1978) to meet the climate modelling requirement. In order to enforce conservation and balance the energy conversion terms it is necessary to advect all the fields with a three-dimensional velocity field consistent with the continuity equation. In the original split-explicit method this was not satisfied because the vertical advection is carried out in the short timesteps and the horizontal advection in the long timestep. The requirements can be satisfied by instead including the advection of a basic state potential temperature in the short timestep, and all the rest of the advection in a long timestep, using as the advecting velocity the average mass-weighted velocity from the short timesteps. This makes the structure of a scheme similar to a standard semi-implicit scheme. In particular, the basic state potential temperature must be chosen according to the criteria established in Simmons, Hoskins, and Burridge (1978).

Practical implementation of the scheme in a global model requires choices of numerical filtering and smoothing methods consistent with the requirements. Fourier filtering in high latitudes is applied to mass-weighted velocity fields, and to mass-weighted potential temperature and moisture increments, so that conservation properties are retained and the fields are not distorted. Numerical noise is removed with conservative high order filters. The order used depends on the resolution of the model. There has been much recent interest in the use of advection schemes which prevent the development of spurious oscillations, e.g. Williamson and Rasch (1988). The most suitable way of constructing such schemes within the split-explicit framework is that introduced by Roe (1983), where advective increments calculated on the boundaries of control volumes are distributed upwind or downwind according to a suitable limiting criterion. The implementation of one of these is described.

II. A CONSERVATIVE SPLIT EXPLICIT SCHEME

For the purposes of this paper, the scheme is written in Cartesian coordinates. The extension to spherical geometry is straightforward. The equations required to treat moisture are omitted, as are additional small terms included in the model to allow more accurate

\[ u^{t+\delta t} = u^t + \delta t \left( \frac{1}{2} \left( v^{t+\delta t} + v^t \right) - \frac{\delta x}{\Delta x} \right) \]

\[ \frac{1}{\Delta x} \left\{ \left( \frac{\partial}{\partial x} \right) \left[ \frac{P_k}{\frac{1}{2}} \right] \right\} + \frac{\delta x}{\Delta x} \right] \]

\[ \delta p_k \left( \frac{\Pi_{k-1/2} P_{k-1/2} - \Pi_{k+1/2} P_{k+1/2}}{\left( k+1 \right) \Delta p_k} \right) \]

at each level \( k \), with a similar equation for \( v \). \( f \) is the Coriolis parameter, \( c_p \) the specific heat of air at constant pressure, \( \kappa = R/c_p \), where \( R \) is the gas constant, \( \varphi \) is the geopotential, and \( \Pi \) is the Exner function \((p/p_0)^{\kappa} \) with \( p_0 \), a reference pressure. The standard finite difference averaging notation is used. The hydrostatic equation is approximated by

\[ \varphi_k = \varphi_k - \sum_{m=1}^{\infty} \frac{\partial}{\partial x} \left( \Pi_{k+1/2} - \Pi_{k-1/2} \right) \]

\[ \frac{\delta x}{\Delta x} \]

\[ \delta p_k \left( \frac{\Pi_{k-1/2} P_{k-1/2} - \Pi_{k+1/2} P_{k+1/2}}{\left( k+1 \right) \Delta p_k} \right) \]

The special form of the last term is chosen to ensure angular momentum conservation.

The second half of the forward-backward step solves

\[ p_*^{t+\delta t} = p_*^t + \delta t \sum_{m=1}^{n} D_m^{t+\delta t} \]

\[ \frac{\delta x}{\Delta x} \]

\[ \frac{\delta p_k^{t+\delta t}}{\partial x} \left( \frac{\Pi_{k-1/2} P_{k-1/2} - \Pi_{k+1/2} P_{k+1/2}}{\left( k+1 \right) \Delta p_k} \right) \]

where \( \Pi_* \) is a basic state profile of \( \sigma \) calculated from an isothermal basic state with temperature 300 K and surface pressure 100000 pa. as used in standard semi-implicit models. The theory behind the choice is exactly that in Simmons, Hoskins, and Burridge (1978). The advection scheme used here is described in detail by Cullen and Derber (1981).
\[
(U_k^y, V_k^y) = (u_k \Delta p_k, v_k \Delta p_k \cos \phi)
\]
\[
\frac{\partial E_{k+1/2}}{\partial \eta} = \eta \frac{\partial E_{k+1/2}}{\partial \eta}
\]
as saved from the short timesteps. The finite difference equations for the first advection step are then
\[
\partial_t U_k = \partial_{x_k} \left[ (U_k \delta \phi + V_k \delta \phi) \right] - \frac{\partial}{\partial x_k} \left[ (U_k \delta \phi + V_k \delta \phi) \right]
\]
Similar equations, with the different spatial averaging appropriate for the position of the variables on the B grid, are used to advect the other variables. The second step is written
\[
\partial_t U_k = \partial_{x_k} \left[ \delta \phi_x \partial x_k \right] - \frac{\partial}{\partial x_k} \left[ \delta \phi_x \partial x_k \right]
\]
This form is chosen to ensure conservation under time differencing.

III. OSCILLATION-FREE ADVECTION SCHEMES

The most natural way of incorporating oscillation free advection schemes into this structure, while retaining conservation is that of Roe (1983). The standard advection scheme set out above can be considered as two stages. First calculate advective increments of the form
\[
\frac{U_k \delta \phi}{\Delta x_k}
\]
at grid box boundaries. The second step is to apply these increments to adjacent grid points. This is done in the standard scheme by applying half the increment to the point on either side of the boundary. Upwind schemes can be generated by applying the whole increment to the point downstream of the boundary. Accurate oscillation free schemes are obtained by using a more accurate redistribution algorithm, several of which are set out in Roe's paper. The resulting schemes are algebraically equivalent to those obtained by starting from the conservation law form of the equations and applying a flux limiter. However, the use of short timesteps for the continuity equation and long timesteps for the advection makes it harder to set out the schemes in that form.

The schemes can then be applied to a three-dimensional model on the sphere by combining three one-dimensional schemes. Thus the redistribution is calculated separately for each coordinate direction. Fourier filtering destroys the oscillation-free property. At high latitudes, the east west sweep must be repeated several times to avoid compromising the model timestep.

REFERENCES

Gadd, A.J. 1978
Roe, P.L. 1983
Simmons, A.J. and Burridge, D.M. 1981
Simmons, A.J., Hoskins, B.J. and Burridge, D.M. 1978
Williamson, D.L. and Rasch, P.J. 1988
COMPUTATIONAL DISPERSION PROPERTIES OF VERTICAL GRIDS FOR ATMOSPHERIC MODELS

Michael S. Fox-Rabinovitz  
Laboratory for Atmospheres  
NASA/Goddard Space Flight Center  
Greenbelt, MD 20771, USA

ABSTRACT

Computational dispersion properties of different vertical grids, namely for regular, Lorenz, Charney-Phillips, and a new class of time-staggered grids have been studied using a linear baroclinic hydrostatic atmospheric model and intercompared in terms of frequency, phase and group velocity characteristics. It is shown that the widely used Lorenz grid, and also the Charney-Phillips grid as well as the new time-staggered versions of these grids have dispersion properties corresponding to a regular grid with twice the vertical resolution. They are computationally efficient due to enhanced effective vertical resolution. The scale ranges for which group velocities have the wrong sign are pointed out. The influence of higher (4th) order vertical approximation has been investigated and found to be relatively insignificant. The practical applicability of time-staggered vertical grids has been tested and proven effective within full time-space staggered grids for an experimental PE baroclinic model.

I. INTRODUCTION

The most widely used vertical grid is the Lorenz (1960) type staggered (or L-grid) which carries horizontal velocities and temperatures at the same levels, and vertical velocities at the intermediate levels. The advantage of the Lorenz grid is in its easy maintenance of conservation laws. The Charney-Phillips (1953) type staggered grid, or CP-grid carries vertical velocities and temperatures at the same levels and horizontal velocities at the intermediate levels. The advantage of this grid is its easy maintenance of integral constraints for a quasi-geostrophic flow.

New time-staggered versions of these vertically staggered grids, namely the time staggered L grid or the LTS grid, and time staggered CP grid or the CPT grids are introduced. All staggered grids are computationally efficient due to enhanced effective vertical resolution compared to that of a regular (unstaggered) grid. Their dispersion properties are intercompared both with each other and against that of an analytical case in a manner similar to that used by Mesinger and Arakawa (1976) and Fox-Rabinovitz (1991) for horizontal grids.

II. THE DIFFERENTIAL CASE

Let us consider a linear baroclinic PE hydrostatic atmospheric model in \( \zeta = \beta \phi \) coordinate system.

\[
\begin{align*}
\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + \frac{\partial \phi}{\partial x} &= 0, \\
\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + \frac{\partial \phi}{\partial y} &= 0, \\
\frac{\partial^2 \phi}{\partial \zeta^2} + c^2 \Omega &= 0, \\
\end{align*}
\]

where \( u, v, \Omega = \frac{\partial \phi}{\partial t} \) are velocity components, \( c^2 \) = const., \( \phi \) geopotential, \( f \) constant Coriolis parameter. The solution has the form:

\[
F = F \exp \left[ i (k x + m y + r \zeta - vt) \right],
\]

where \( k, m, r \) are wave numbers, \( v \) frequency. The dispersion relationship and horizontal and vertical group velocity components (HGVC and VGVC) are as follows (for \( k = m \)):

\[
v = B; \quad \text{HGVC} = \frac{\partial v}{\partial k} = 2B^{-1} c^2 K r^2 > 0;
\]

\[
\text{VGVC} = \frac{\partial v}{\partial \zeta} = -2B^{-1} c^2 k^2 r^2 < 0; \quad (3)
\]

These characteristics are used for comparisons of different grid dispersion properties. Note that the most important characteristic is the appropriate sign of group velocity components, namely HGVC should be positive and VGVC negative for all resolved scales.

III. VERTICALLY AND TIME-VERTICALLY STAGGERED GRIDS

The same dispersion characteristics are obtained for the system (1) approximated with a central difference scheme (CDS) for different vertically and time-vertically staggered grids, and compared with each other, and against the differential case (3). For simplicity a regular, or Arakawa A grid is used for horizontal approximations. The vertical grids considered are presented in Figs. 1-5.

We considered first what may be achieved by introducing 4th order CDS in the vertical instead of the 2nd order CDS. The sign of both VGVC and HGVC are wrong for both schemes in certain scale ranges, although for the 4th order CDS they are slightly smaller. Namely, for the 2nd order CDS the aforementioned signs are wrong for scales smaller than \( L < 4 \Delta \zeta \), whereas for the 4th order CDS the signs are wrong for scales \( L < -3.5 \Delta \zeta \) which is quite comparable to the 2nd orders result. Therefore, the dispersion properties cannot be significantly improved by using the higher (4th) order approximation in the vertical with a regular (unstaggered) grid. The most significant feature of all vertically and time-vertically staggered grids considered is that they have definitely better dispersion characteristics than that of a regular (unstaggered) vertical grid. Both HGVC and VGVC have an appropriate sign for all resolvable scales \( L < 2 \Delta \zeta \) due to the higher effective vertical resolution. In general all staggered grids have dispersion properties which correspond to or are very close to those of a regular grid with twice the vertical resolution.

Note that semi-implicit or economical explicit schemes may be complemented with time-vertically staggered grids and the dispersion properties for the case are identical with those for a regular grid with doubled resolution. The examples of VGVCs for different vertical grids are presented in Fig. 6.

Note also that the upper and lower levels for time-vertically staggered grids may be the same for the adjacent time steps which is convenient for orography incorporation and upper boundary condition.

IV. EXPERIMENTS WITH TIME-VERTICALLY STAGGERED GRIDS

The time-vertically staggered grids have been combined with time-horizontally staggered grids within a full 4-D staggered grid approach. The experimental PE baroclinic model with the grid is tested and found to be computationally efficient. The results of regional and storm forecasts are encouraging.
REFERENCES


Fig. 1. A regular (unstaggered) vertical grid. F stands for model variables.

Fig. 2. The vertically staggered Charney-Phillips grid (1953), or CP grid.

Fig. 3. The vertically staggered Lorenz grid (1960), or L grid.

Fig. 4. The time-vertically staggered CP grid, or CPTS grid.

Fig. 5. The time-vertically staggered L grid, or LTS grid.

Fig. 6. Vertical group velocity components for the differential case (a); a regular vertical grid with the 2nd order CDS (b); vertically staggered L and CP grids (c); and time-vertically staggered LTS and CPTS grids (d).
I. INTRODUCTION

One aspect of weather and/or climate prediction which limits our capabilities is associated with computational errors arising from the numerical representation of the relevant differential equations describing the prediction system. Of the representational methods currently favored, global function expansion (spectral), finite differencing in space of finite element functions, each has its own limitations, but the interrelationship amongst them has not been carefully discussed. Issues such as aliasing errors and computing speed compete when a representational choice is to be made. Whereas the spectral method is preferred for global calculations, it is not suitable for regional calculations with complex boundary conditions.

It is the intent of this report to present a procedure which, when applied in the spatial domain, could yield results which are equivalent to those derived from the spectral domain. If the equivalence can be demonstrated in application to general models, the choice of numerical representation would be one of preference or convenience rather than one associated with forecast quality. We shall lean heavily on linear theory to present our case.

II. PROCEDURE

We choose for our analysis the model developed by Kalney, et al. (1977). This model is representative of the models in current use as prediction systems and is further available in 4th order difference form at GSFC/QLA, if and when we need a comprehensive model to test our results. Since the model is global, it would also be possible to compare a spectral version with the existing finite-difference form. To understand the structure of the prediction equations, we introduced the model without forcing about a state of rest.

The resulting system yields to a separation in variables such that the horizontal dependence may be solved for independently of the vertical dependence, the separation constants are derived from the solution to the equations in the vertical dimension and are denoted as equivalent depths with corresponding eigenvectors which have been carefully studied by Baer and Ji (1989). If Fourier series are substituted for the longitudinal dependence, the shallow water equations, dependent only on latitude, result and have analytic solutions. These solutions are known as Hough functions and depend on the equivalent depth and the planetary wave number. There are an unlimited number of these functions which satisfy the equations, and a finite set would reflect modal truncation.

To understand how a finite-difference of finite element representation of the shallow water equations relates to the spectral (Hough) solutions, we have evaluated the latitudinally dependent equations on an equally spaced grid of points in latitude, and using both the 4th order differencing scheme and a second order finite-element scheme applied to the general model, we have developed two discrete systems, each of which yield a set of eigenvectors and eigenvalues, then number and character depending on the number of grid points selected. Since the true solutions are known (Hough functions), these eigenvectors can be compared to the Hough solutions and identified as true solutions (those which separate) or as false solutions (those which do not separate and consequently represent computational errors). If a one-to-one comparison exists and only the true eigenmodes are used to solve the linear system on the grid, then the results of the discrete representation would yield the exact solutions.

Should one consider a different model, or even the same one which was not valid globally, the exact solutions would not be apparent. Thus an alternate procedure is needed to distinguish the true modes from the false ones. We have done this by introducing the shooting method. Simply described, this method utilizes the eigenvalue for a given spatial eigenfunction of the discrete equations, starts at one boundary and integrates the equation point by point to the other boundary. If the other boundary condition is thereby met, the eigenvalue and its associated vector satisfy the equation. If the boundary condition is not met, an adjustment to the eigenvalue is made and the process is repeated. If the solution converges to the boundary condition with minor adjustments of the eigenvalue, the numerical solution is probably true. If it diverges, the solution is false, i.e., no such vector satisfies the original equation. We compared the true and false solutions so identified to the Hough modes, which are by definition the true modes. By this procedure, we can identify the equivalence of a discrete model to a spectral model.

Finally, we have projected observational data fields onto a set of Hough functions and on the corresponding sets of eigenvectors derived from the discrete systems. Since the projections are done independently, some measure of amplitude loss which comes from discarding false modes is made apparent and is available to assess the applicability of the procedure.

III. EXPERIMENTS

The model equations discussed above, essentially the shallow water equations, were transformed to discrete systems by assuming 4th order differencing over an equally spaced latitudinal grid and 1st order finite element representation on the same grid. In both cases, the grids used were 5° latitude and 2°.5° latitude. The corresponding longitude representation was for a continuous Fourier series and a fourth order differencing with the appropriate grid increment to allow for a maximum of 20 planetary waves.

The equations representing each differencing scheme were solved as a matrix problem and yielded eigenvectors and eigenvalues, their number appropriate to the number of grid points used. For example, if the grid increment was 5°, 105 vectors evolved. These vectors were calculated for each planetary wave, latitude, and for the various equivalent depths. Assuming nine vertical levels, we evaluated each of the nine equivalent depths. The minimum number of Hough modes appropriate to a given truncation was established by identifying that set of Hough modes which were included amongst the eigenvalue set of the discrete system, no more or no less.

The shooting method was applied to each set of eigenvectors of the discrete representation for each choice of parameters and each numerical technique. Thus a set of true and false modes was determined for each experiment and was available for comparison with the associated Hough mode solutions.

Finally, several data fields were projected onto the eigenvectors derived from the various experiments, as well as onto the corresponding Hough modes, and the relative amplitude of the data in the true modes as contrasted to the false modes was assessed.
IV. RESULTS

The experiments outlined above have resulted in the following observations.

(a) Sensitivity to truncation in longitude results in variations observed only on the shortest planetary scales, which represent a 2Δλ increment. This confirms many previous observations.

(b) The distribution of modes between Rossby and gravity evolving from the solutions to the discrete (truncated) systems conforms to the expectations for the true (Hough) solutions; i.e., two-thirds of the solutions are gravitational and one-third are Rossby. However, some of the Rossby modes calculated are false because they propagate to the east. Almost half of the Rossby modes calculated for the 4th order system fall into this category, whereas considerably fewer have this property for the finite-element system.

(c) The shooting method works well and as expected. The modes which are defined as true by the shooting method are also Hough modes whereas the false modes are not found amongst the Hough mode solutions.

(d) A direct relationship has been found between the finite-difference increment selected and spectral truncation. Based on item (c) above, this correspondence can be calculated from either the number of true modes found by shooting or the number of relevant Hough modes determined. The results are sensitive both to the planetary wave number and to the vertical mode. Moreover, the response of the Rossby and gravity modes differ. We have tested these results for both ΔΔλ = 5° and ΔΔφ = 2.5°, and the results are consistent. Consider the experiment for ΔΔλ = 5° with 4th order differencing. For the external vertical mode and the longest planetary wave (m=1), almost one-half of the Rossby modes are true and 45 percent of the gravity modes are true. This translates roughly to 16 Hough modes for each set (Rossby, gravity-east and gravity-west).

However, for the longest planetary wave and the sixth internal mode, only twelve Rossby Hough modes are true and only 14% gravity modes are true. For the external mode at planetary wave twenty, twelve Rossby modes are true but only five gravity modes are true. The implications here are clearly that spectral truncation and the corresponding finite-difference truncation are highly scale dependent.

(e) The impact of false modes on the integration of a nonlinear system may be assessed by considering their involvement in the energy exchange process. We have investigated the impact by projection of several data fields onto all the modes of a discrete system. As an example, consider the rotational kinetic energy of the external mode expressed by the modes of the 4th order differencing system. If we compute the ratio of energy in the true modes to the energy in all modes, we note that for the Rossby modes, this ratio decays from near 95 percent for the largest planetary wave to near 80 percent at wave number twenty. By contrast the ratio is near 40 percent for the largest planetary wave for gravity waves, and gradually increases to near 50 percent by wave twenty. Since the energy in both groups decays rapidly with wave number, and furthermore since the Rossby energy is an order of magnitude larger, the total energy in this vertical mode is described to better than 90 percent by the true modes. This result is similarly noted for the larger internal vertical modes.

V. CONCLUSIONS

The results of our calculations indicate that a correspondence can be made between spectral and grid truncation, and based on our analysis with a relatively simple model, it is highly scale dependent. Whereas for the longest planetary waves and the external vertical mode the conventional concept that one wave for each two grid points applies, very few spectral components represent many grid points for short planetary waves and internal vertical modes.

Truncation decisions can be made by considering the distribution of observed variables in terms of the appropriate eigenvectors of the system to be predicted. Once a decision on the number of false modes which can be tolerated is made, the appropriate truncation can be chosen and a grid or spectral truncation can be selected. Aliasing errors during nonlinear integrations can be reduced by filtering the initial data of the unwanted false modes. Because those modes exist in the prediction system, their amplitude will grow in time due to nonlinear interactions. To maintain control of these unwanted modes, they may be filtered periodically during the integration cycle. This procedure is clearly not needed in a spectral model.

Tests with a shallow water nonlinear model have been made and indicate how fast false modes grow with time during integration. Such calculations provide a time scale for filtering of the false modes and also give an indication of how successfully a filtered model can provide successful forecasts.

REFERENCES


Abstract - A method for expanding dependent variables in numerical models of the global atmosphere is discussed. The method is shown to be compatible with spectral methods and a procedure for converting spectral coefficients to polynomial coefficients is demonstrated. The polynomial method proves more efficient on a scalar computer and has potential to be more efficient than traditional spectral algorithms.

I. INTRODUCTION

During the development of spectral models of the atmosphere in the 1970s, most researchers realized that a major difficulty with spectral methods is the evaluation of Legendre functions. Although swift and accurate methods exist for the computation of Legendre functions, it was not feasible to store all the necessary values of the functions and to access them frequently during numerical integration because of the drag it normally imparts on computer processing.

Several researchers have, therefore, looked into the possibility of expressing the Legendre functions in terms of Fourier expansions divided by some power of \((1-\mu^2)^{1/2}\), i.e., \(\cos \phi\) where \(\mu = \sin \phi\) and \(\phi\) is latitude. Herilees (1973) suggested that the Legendre function be expressed as \(P_n(m/2)\) where \(P_n(m/2)\) is an expansion of \(\sin \mu\), \(\mu\) ranges from 0 to \(\pi\), \(\phi\) being the zonal wave number. Orszag (1974) suggested that the function be represented as \(G_n(\mu)(1-\mu^2)^{n/2}\) where \(G_n(\mu)\) is also an expansion of \(\sin \mu\) while \(z\) can be either 1 or 0 depending on whether \(n\) is even or odd. Yee (1980) extended the argument so that the Legendre function could be expressed completely as a Fourier series, where cosine terms are kept for even functions and sine terms are kept for odd functions.

In this study, we offer an alternative more compatible with Legendre functions and hence with spectral models. It involves shifting from expansions in terms of Legendre functions to expansions in polynomials of \(\mu\), while keeping the representation of the fields spectral.

II. SPECTRAL EXPANSIONS

To evaluate a given variable, \(A\), which is a function of latitude, \(\phi\), and longitude, \(\lambda\), from its spectral coefficients, \(a_j^n\), we execute the sum

\[ A(\phi, \lambda) = \sum_{n=-N}^{N} \sum_{m=|n|}^{N} a_j^n p_j^n(\phi) e^{i n \lambda} \]  

(1)

where \(p_j^n(\phi)\) are the associated Legendre functions. The Legendre functions themselves are defined as

\[ p_j^n(\phi) = (1-\cos^2 \phi)^{n/2} \sum_{j=0}^{n} \frac{(-1)^{n-j}}{j!} \frac{(n-j)!}{j!(n-j)!} \cos^j \phi \]

and the unknown \(a_j^n\), one can easily solve for the coefficients \(a_j^n\) need be derived only once and stored for future use. Because \(P_n^m(\phi)\) is even for even \(n\) and odd when \(n\) is odd, only of the order of \((N+1)/4\) coefficients need be saved for each \(m\) under Rombergian truncation, where \(N\) represents the largest meridional wave number of the expression.

There are various ways to derive the \(a_j^n\), but there is no space here to expound upon them. Once they are known it is possible to substitute into (1) to derive the Fourier coefficients, \(A_j^n(\phi)\) of \(A(\phi, \lambda)\), given as

\[ A_j^n(\phi) = \frac{1}{\sqrt{2\pi}} \int_0^{2\pi} d\lambda A(\phi, \lambda) e^{-i m \lambda} \]

Rearranging terms gives

\[ A_j^n(\phi) = \frac{1}{\sqrt{2\pi}} \sum_{m=-N}^{N} \sum_{n=|m|}^{N} a_j^n e^{i m \lambda} \]

Thus, given \(A_j^n\) and the coefficients \(\sigma_j^n\), we can determine the coefficients \(a_j^n\) and evaluate the variable in terms of powers of \(\mu\). The summation leading to \(C_j^n\) is a very swift process, especially on vector computers and, of course, the summation of powers of \(\mu\) in (2) is much quicker than the evaluation of Legendre functions.

III. INVERSE TRANSFORMS

In the course of most spectral model processing, it is necessary to obtain the spectral coefficients from the values of the parameters, e.g., for non-linear terms this is normally performed at every time step. The polynomial approach can also furnish the same spectral coefficients but without explicitly calculating the Legendre functions and performing the related Gaussian integration. As with the spectral method, the function \(A(\phi, \lambda)\) is decomposed by Fourier transform to obtain the Fourier coefficients at \(N+1\) latitudes. We then divide \(A_j^n(\phi)\) by \((1-\mu^2)^{n/2}\) and evaluate \(C_j^n(\phi)\), provided \(\phi \neq \pi/2\). We now solve the system of \(N+1\) linear equations given by (2) to obtain the coefficients \(\sigma_j^n\), \(j=0, ..., N\). By inverting the relationship between the known \(C_j^n\) and \(a_j^n\), one can easily solve for the

583
spectral coefficients recursively by

\[
A_{n+1} = \frac{C_n}{A_n} + I_n + A_{n-1} \frac{A^n_{n+2}}{A^n_{n+1}} + \ldots + \frac{C_{n-k}}{A_n} + A_{n-k+1} \frac{A^n_{n+2}}{A^n_{n+1}} \frac{A^n_{n+3}}{A^n_{n+2}} + \ldots + \frac{A^n_{n+r}}{A_n} \frac{A^n_{n+r+1}}{A^n_{n+r}} \frac{A^n_{n+r+2}}{A^n_{n+r+1}} \ldots \frac{A^n_{n+r+k}}{A_n} \frac{A^n_{n+r+k+1}}{A^n_{n+r+k}}
\]

for each \( n \). Inverting an upper triangular matrix would be the equivalent operation and may prove swifter.

IV. CONCLUSIONS

In experiments executed on a CDC Cyber 750 (an older, scalar machine), the expansion of temperature from spectral coefficients with rhomboidal truncation of 30 using the standard Legendre function approach over 60 latitudes took 1.472s of CPU time as opposed to .525s for the polynomial method. The inverse transform when performed by Legendre functions and Gaussian integration took 33s to evaluate the spectral coefficients for temperature, moisture, and velocity at 12 levels but only 84s with the polynomial method. Tests on faster vector processing machines have not been performed to date, but savings in time may be expected there as well. It must be remembered, as well, that the proposed method here is completely spectral and is therefore completely compatible with other spectral models.

There are some drawbacks to the method, however, which must enter in its evaluation. First, expressing the Laplacian is much simpler with Legendre functions than with polynomials, but the polynomial expression is still tenable. Second, it is necessary to store the coefficients of the powers of the sines that generate the Legendre functions, but this, again, may not prove too difficult especially with large mainframes. Third, accuracy near the poles may present a problem, but this affects Legendre functions as well. All in all, the polynomial approach may serve as a beneficial substitute for conventional spectral methods.

V. ACKNOWLEDGEMENT

Research and preparation of this document have been funded by the U.S. Air Force under Contract No. F19628-88-C-0072.

REFERENCES


I. Introduction

There are principally two types of waves represented in current operational atmospheric models: Rossby-waves and gravity waves. The former typically propagate at speeds of about 30 m/s and the latter at speeds up to 300 m/s. In the early days of numerical weather prediction (NWP), because of their diverged wavelengths and high speeds, gravity waves were treated with respect but also with awe. The common wisdom was to filter out the shorter ones and treat gravity waves as a kind of "shallow" fluid on an f-plane. We can write  

\[
\begin{align*}
(hu)_t &= -(uh)_x + f(hv) - g(h^2/2)_x + \nu(hu)_{xx} \\
(hv)_t &= -(uv)_x - f(hu) + \nu(hv)_{xx} \\
(h)_t &= -(hu)_x 
\end{align*}
\]

where \( g \) is the acceleration of gravity, \( f \) is the Coriolis parameter, and \( \nu \) is an eddy viscosity. The height-weighted velocities, \( hu \) and \( hv \), are analogous to the pressure-weighted velocities used as prognostic variables in many mesoscale hydrostatic models. The continuity equation, (3), is analogous to the pressure tendency equation in a hydrostatic model.

We nondimensionalize the equations using the undisturbed height, \( H \), as a scale. Thus, letting primes denote the nondimensional quantities, we have

\[
\begin{align*}
\frac{h}{H} &= \frac{h'}{H} \\
u &= \sqrt{\frac{ghH}{\nu}} \frac{u'}{H} \\
v &= \sqrt{\frac{ghH}{\nu}} \frac{v'}{H} \\
\frac{t}{T} &= \frac{t'}{T} \\
\frac{x}{L} &= \frac{x'}{L} \\
\frac{f}{fH} &= \frac{f'}{fH} \\
\frac{h}{H} &= \frac{h'}{H} \\
\frac{h^2/2}{H^2} &= \frac{(hH/2)'}{H} \\
\end{align*}
\]

Then the nondimensional equations are (dropping primes)

\[
\begin{align*}
(hu)_t &= -(uh)_x + f(hv) - (h^2/2)_x + \frac{K(hu)_{xx}}{H} \\
(hv)_t &= -(uv)_x - f(hu) + \frac{K(hv)_{xx}}{H} \\
(h)_t &= -(hu)_x 
\end{align*}
\]

Note that \( K \) may now be regarded as an inverse Reynolds number and that we have separated the height-weighted divergence term in (7) into perturbed and undisturbed parts. Following the choices of Seitter, we let \( K = 0.0078, f = 10^8 s^{-1}, H = 8 kn \) (approximately the density scale height), which yields an external gravity wave speed of \( \sqrt{gh} = 280 m/s \).

III. The Dual Grid Method

The two key issues in implementing the dual resolution method are: (a) the identification of the model forcing terms which contribute to shorter gravity waves and thus warrant higher resolutions, (b) the design of an interchange information transfer procedure between the finer grid \( (d_0(x,t)) \) and the coarser grid \( (d_1(x,t)) \), where \( d_0 \) and \( d_1 \) denote increments in grid points \( d_0 \) and \( d_1 \), respectively. The identification problem is not always easy. In one case, however, it is known that the balance between the pressure gradient and the linear divergence terms are largely responsible for the gravity waves. Although the role of the viscous terms is to simulate damping, we also discretize them at a higher resolution on the grids that they represent smaller scale phenomena and that in more realistic models, \( K \) is a function of the flow field itself. For internal transports, we simply use discrete Fourier smoothing and interpolation.
Thus given initial conditions on $d$, we

1. smooth values of dependent variables on $d$ to get smooth values on $D$;
2. compute longwave tendencies $R(S)$ on $D$;
3. interpolate $R(S)$ to get $R(S)$ on $d$;
4. compute gravity wave tendencies $g(S)$ on $d$;
5. time-integrate on $d$, using $R(S)$ from (3) and $g(S)$ from (4) above;
6. repeat steps (4) and (5) for $(At/St)$ times;
7. repeat steps (1) through (6).

The main point here is that time integration is done on the finer grid at time intervals $At$, but with the longwave tendencies computed only on the $D$ grid. Another way of looking at it is that while longwave tendencies are computed less frequently on a coarser grid, they are interpolated onto the finer grid and then fed into the total tendencies at the smaller $At$ interval through step (5). It should be noted that step (1) ensures finer grid/coarser grid interactions and that this selective adaptation of grid resolutions to the characteristic scales of the modeled phenomena should be done both spatially and temporally. Although our multi-resolution approach, in which the sizes of the time-steps are tailored to wavelengths and wave-speeds, enables us to use the computationally much simpler explicit time-scheme, for the studies reported here we have incorporated Robert's semi-implicit time scheme for pedagogical purposes.

IV Preliminary Results

The procedure given in Section III has been tested with the shallow water model. For comparison purposes, model equations were solved in two different ways. In one, we simply used a centered-difference, explicit time scheme with a weak time filter (filtering coefficient $\alpha = 0.02$) to couple even/odd time steps in the conventional manner. In the other, we solved a set of equations derived from (5), (6) and (7) using the dual-grid semi-implicit method. Periodic boundary conditions were used for a 38,400 km domain which contains 240 finer mesh points ($\Delta x = 160$ km), but contains 15, 30, 60, and 240 mesh points, respectively, for various coarser grids. Time-step $At$ was set at 60 seconds, with the coarse grid $At = 2At$. A small $At$ was used deliberately, although a 300 second $At$ has been used in some test cases for time-integrations up to 5 days.

Initially the fluid was assumed to be at rest. A sudden disturbance one-tenth the size of the domain was then imposed on $h$ at the center of the domain, as shown schematically by the upper solid curve in Fig. 1. (Only half of the domain is shown.) The other curves in Fig. 1 show the height distributions of the free surface at model times 4, 8, 16, and 24 hours. We have displaced the curves vertically for clarity in the display.) We see that the free surface waves propagate both up and down stream with diminishing amplitudes during the first 24 hours in the model. In fact, the primary waves shown for hour-24 are retrograde waves originating at the center of the domain. In 24 hours, they have propagated slightly more than half of the domain. Fig. 2 shows the energy spectra of the height fields for the longest 25 waves at $t = 0$, and at 24 hours as given by the two methods of solution. Initially, the energy spectrum is relatively smooth, with a single peak at wave numbers 15 and 16. At 24 hours, a bimodal distribution has developed in both model solutions, apparently due to the rapid removal of energy at the initial spectral peak. Furthermore, even in the cases shown (dual grid solution with $\Delta x = 166\Delta x$), where the solutions diverge the most and where the energy spectra seem to be off-set by one or two wavenumbers, on the whole, the two solutions possess remarkably similar spectral characteristics.
Optimal Instability of Shear Flows in the Initial Value Problem

Enda O'Brien
University of Miami
Division of Meteorology and Physical Oceanography
4600 Rickenbacker Causeway
Miami, FL 33149, U.S.A.
Tel. (305) 361-4032

Abstract A variational principle is used to find those perturbation structures which have the fastest divergence from a basic state in linearized shear flows, and for finding the initial rate of divergence. The technique requires the existence of a positive-definite quantity (e.g., energy or potential enstrophy) whose growth rate can be expressed as a function of the instantaneous perturbation structure. The problem reduces to a fourth-order nonlinear ordinary differential equation for the instantaneous perturbation structure. The problem reduces to a fourth-order nonlinear ordinary differential equation for the structure function, which can be solved using various iterative numerical techniques. Green’s model of baroclinic instability [1] is presented as an illustration of the method.

1 Introduction

A large class of waves in shear flows have instantaneous growth rates significantly larger than normal mode growth rates [2]. A consequence of this is that error growth in numerical forecast models can be faster than predicted by calculations of the first Lyapunov exponent (or growth rate of the most unstable normal mode). Questions which then arise are: is there a bound on the instability of these non-modal disturbances? If so, what is it and what is the structure which attains this bound?

In order to define a measure of instability, we seek a norm such that the perturbation growth rate in that norm can be expressed entirely as a function of the perturbation structure and the basic state flow. The perturbation structure can then be varied in order to maximize the growth rate.

2 Green’s model

Consider a basic state wind \( U(x) \), in a Boussinesq fluid with constant stratification on a \( \beta \)-plane, confined between rigid surfaces at \( z = 0 \) and \( z = 1 \). The governing equations are the linearized nondimensional quasi-geostrophic potential vorticity equation, along with the boundary conditions

\[
q_e = -U q_e - v Q_v, \quad 0 < z < 1 \quad (1)
\]

\[
\psi_e = -U \psi_e + v U_s, \quad z = 1 \quad (2)
\]

\[
\psi_\ast = -U \psi_\ast + v U_s - \gamma \nabla^2 \psi, \quad z = 0. \quad (3)
\]

Here lowercase letters represent perturbation quantities, while uppercase letters refer to basic state quantities. The basic state vorticity gradient is \( Q_v \) defined by:

\[
Q_v = \beta - U_{z}, \quad (4)
\]

The dimensionless parameter \( \gamma \) represents Ekman pumping at the ground. The perturbation streamfunction \( \psi(x, y, z, t) \) is related to the wind components by \( u = -\psi_y, \quad v = \psi_x \), and to the potential vorticity \( q \) by

\[
q = \nabla^2 \psi + \psi_{xx}. \quad (5)
\]

An energy equation is obtained by multiplying (1) by \( -\psi \) and proceeding in the usual way. We obtain

\[
dE \over dt = - \int_0^1 \overline{vq} U ds - r(\nabla \psi^2)_{t=0}. \quad (6)
\]

Overbars represent a horizontal domain average. The total energy \( E \) is defined by:

\[
E = 0.5 \int_0^1 (\nabla^2 \psi)^2 + (\psi_\ast)^2 ds. \quad (7)
\]

The instantaneous energy growth rate \( \sigma_E \) is now defined as \( E^{-1}dE/dt \).

An equation for the perturbation potential enstrophy \( H \) is obtained by multiplying (1) by \( q \) and proceeding as for the energy equation. The potential enstrophy growth rate \( \sigma_H \) is defined as \( H^{-1}dH/dt \). For an arbitrary perturbation \( q \) there is no reason to expect \( \sigma_E \) and \( \sigma_H \) to be equal.

3 Constraints

Consider first the inviscid case (i.e., \( r = 0 \)). Where \( Q_v \neq 0 \), we can multiply (1) by \( q \), average horizontally, divide across by \( Q_v \), and integrate vertically to obtain

\[
\frac{dA}{dt} = - \int_0^1 \overline{vq} ds \equiv 0, \quad (8)
\]

where the wave action (or pseudomomentum) \( A \) is defined (formally) by

\[
A = 0.5 \int_0^1 (q q)^2 dz. \quad (9)
\]

Where \( Q_v \) vanishes the definition of \( A \) in (9) can be modified straightforwardly. Eq. (8) implies that \( A \) is a constant. We set \( A = 0 \) since this is the only value for which \( A \) is independent of the perturbation amplitude, which is arbitrary.

When surface friction is included, the appropriate constraint becomes \( \sigma_A = \sigma_X \), where \( \sigma_X \) is the growth rate in some norm (e.g., the energy or potential enstrophy norm), and the wave-action growth rate \( \sigma_A \) is defined as \( A^{-1}dA/dt \).

4 The variational principle

The problem now is to maximize \( \sigma_E \) (or \( \sigma_H \)) subject to the constraints derived above. In the inviscid case, we render stationary the functional \( L \), where

\[
L = \sigma_E - \lambda A. \quad (10)
\]

In the case with friction, (10) is replaced by

\[
L = \sigma_E - \lambda (\sigma_E - \sigma_A). \quad (11)
\]

For optimal potential enstrophy growth rates \( \sigma_E \) is replaced by \( \sigma_H \) in (10) and (11). Now let \( \psi(x, y, z, t) = \psi(x, t) \overline{U(z)\exp(ikb)} + cc. \) From here on the method is exactly analogous to the derivation of Lagrange’s equations from a variational principle [3].

The same procedure is followed for all cases. For optimal energy growth in the inviscid case, we obtain a fourth order nonlinear ordinary differential equation as a structure equation for \( \psi(x) \) (henceforth dropping the "hat"). For \( U_s = const \) (the classical Green’s model) this equation reduces to:

\[
\psi_{\ast\ast} = -2(2K^2 + \gamma \sigma_E) \psi_{\ast\ast} + 2\gamma k U_s \psi_\ast + (K^2 + \gamma \sigma_E K^4) \psi = 0, \quad (12)
\]

where \( \gamma = Q_v/(\lambda E) \) and \( K^2 = k^2 + l^2 \).

Boundary conditions may be obtained for (12) in exactly the same way that the boundary conditions (2) and (3) are obtained for (1).

This yields:

\[
\psi_z = (i k U_s/\sigma_E) \psi, \quad z = 0; \quad z = 1. \quad (13)
\]

Since growth rate is independent of the wave’s amplitude and phase, these quantities may be arbitrarily fixed at one level to provide a third condition on (12). A fourth condition is the integral constraint \( A = 0 \), which is necessary to determine the value of the multiplier \( \lambda \).

Assuming that a consistent value of \( \sigma_E \) exists, the general solution to (12) has the form

\[
\psi = \sum_{i=1}^4 A_i \exp(ii\xi), \quad (14)
\]
where \( m_i \) are the roots of a quartic equation and \( A_i \) are (complex) constants determined by the boundary conditions and constraints.

One approach to solving the problem (12) with its boundary and integral conditions is to fix \( \lambda, E \) and \( \sigma_E \) initially at some arbitrary values. The "constants" \( A_i \) in (14) can then be determined. This solution can be used to update \( \sigma_E \) and \( E \) in (12).

Proceeding iteratively in this way, the values of \( \sigma_E \) and \( E \) used in (12) may or may not converge to values consistent with their definitions. Convergence only occurs for isolated values of \( \lambda \).

The approach outlined above was used in solving the Eady problem \( (\beta = 0) \). For the general problem, however, the optimal structure for \( \psi \) was obtained by discretizing in \( z \), thereby reducing (12) to an algebraic problem which was solved using standard routines.

5 Results

In the results presented here, \( \beta = 0.5 \) and the optimization and eigenvalue problems are both solved in a 20-layer domain. Where friction is included, a value of \( \tau = 2 \) is used.

Fig. 1 shows \( \sigma_E, \sigma_H \), and the growth rate for quadratic functions of the most unstable normal mode (referred to as \( \sigma_N \)) as functions of zonal wavenumber \( k \), when \( \sigma_E \) is optimized. The normal mode growth rates \( (\sigma_N) \) show the well-known separation between the so-called Charney modes \( (k > 1.75) \) and the Green modes \( (k < 1.75) \). The curves for optimal \( \sigma_E \) and the corresponding \( \sigma_H \) all decrease monotonically as \( k \) increases and do not have a short-wave cutoff, at least in the inviscid case.

![Energy growth rate optimized](image)

Fig. 1. Growth rate as a function of zonal wavenumber \( k \) \((l = 0)\) for the most unstable normal modes (quadratic growth rate \( \sigma_N \)), the optimal energy growth rate \( (\sigma_E) \), and the corresponding potential enstrophy growth rate \( (\sigma_H) \). Solid curves are from the inviscid case, dashed curves from the case with friction.

By the measure of growth rate in the energy or potential enstrophy norms, optimal structures have been found which are significantly more unstable than normal modes, because of their freedom to grow at different rates in different places. Normal modes, which grow at the same rate everywhere, appear in this perspective as severely-constrained optimal structures. Indeed, optimal perturbations can grow (at least initially) even where there is no normal mode instability at all.

An intuitively appealing feature of the optimally unstable structures is that they vary only slightly in \( N \)-layer models as \( N \) is changed. Normal mode growth rates and structures, on the other hand, can be very sensitive to the value of \( N \). In \( N \)-layer models, \( N \) because of resolved levels falling near critical levels. In our work the concepts of phase speed and critical levels only arise in the specially constrained case of normal modes.

The mathematical approach employed in this study can be applied to a wide range of instability problems, including the instability of initial conditions in forecast models.

6 Conclusions

By the measure of growth rate in the energy or potential enstrophy norms, optimal structures have been found which are significantly more unstable than normal modes, because of their freedom to grow at different rates in different places. Normal modes, which grow at the same rate everywhere, appear in this perspective as severely-constrained optimal structures. Indeed, optimal perturbations can grow (at least initially) even where there is no normal mode instability at all.

An intuitively appealing feature of the optimally unstable structures is that they vary only slightly in \( N \)-layer models as \( N \) is changed. Normal mode growth rates and structures, on the other hand, can be very sensitive to the value of \( N \) in \( N \)-layer models because of resolved levels falling near critical levels. In our work the concepts of phase speed and critical levels only arise in the specially constrained case of normal modes.

The mathematical approach employed in this study can be applied to a wide range of instability problems, including the instability of initial conditions in forecast models.

References

A neural network approach is presented for making short-term predictions on time series. The neural network does better at short term predictions of a chaotic signal than does an optimum autoregressive model. Also the neural network is clearly capable of distinguishing between chaos and additive noise.

I. INTRODUCTION

One of the basic tenets of science is making predictions. If we know previous behavior, how can we predict the future behavior. The approach in many sciences requires two steps; construct a model based on theoretical considerations and use measured data as initial input. Since in many cases the underlying theoretical principles are known, model construction has been and continues to be a primary area of interesting research.

One class of alternative approaches is to build models directly from the available data. For these methods the data, given as a time series, is usually considered a single realization of a continuous random process. This is appropriate when the randomness is a result of complex interactions involving many independent and ultimately irreducible degrees of freedom. Along these lines, linear models have had some success especially in regards to relating cause and effect to physical phenomena, however, their predictive power is limited. The limitation is perhaps related to the inability to model the evolutionary dynamics of the system [1].

In the last decade advances in the theory of dynamical systems have demonstrated the existence of dissipative systems whose trajectories that depict their asymptotic final states are not confined on limit cycles (periodic evolutions) or tori (quasi-periodic evolution) but in submanifolds of the total available phase space which are not topological. These submanifolds are fractal sets and are often called strange attractors. The corresponding dynamical systems are called chaotic systems and their trajectories never repeat. Thus, their evolution is aperiodic but completely deterministic. Because the evolution is aperiodic any "signal" measured from a chaotic dynamical system "looks" quite irregular and exhibits frequency spectra with energy at all wavelengths (broadband spectra) similar to those of random "signals". Another important property of chaotic dynamical systems and their strange attractors is the divergence of initially nearby trajectories. Due to the action of the attractor the evolution of the system from two (or more) nearby initial conditions will soon become quite different. Since the measurement of any initial condition is subjected to some error such a property imposes limits on long-term prediction. Nevertheless, for a short time nearby trajectories may not diverge significantly and thus even though each individual evolution might be quite complex, the knowledge of the dynamics and especially of the structure of the attractor (dimensions, Lyapunov exponents, etc.) may prove beneficial to the art of short-term prediction.

Motivated by the above ideas, very recently a number of techniques for making predictions have been developed to exploit the underlying determinism in complex systems [2,3,4]. The purpose of this paper is to show that neural networks are capable of making short-term predictions on time-series data that are better than an optimum autoregressive model and to show that such a methodology is capable of distinguishing chaos from noise.

II. EXAMPLES

In this section we present two examples showing the effectiveness of using a neural network for making predictions on time-series data. In general neural networks work by iteratively solving for a weight matrix which is used for the inner product that takes inputs to outputs. For time-series prediction, the inputs are taken as lagged values of the discrete time sequence. More details concerning neural networks are given in Rumelhart et al. [5] and Owens and Filkin [6].

The neural network architecture we employ consists of three layers, one input, one hidden, and one output layer (fig. 1).
made further into the future. To do this we make a prediction one step into the future and then use this predicted value as one of the lagged inputs for the next prediction two time steps into the future. Similarly, the prediction at this second time step as well as the previous time step are used as lagged inputs for the next prediction three time steps into the future. Doing this successively allows us to compute the correlation coefficient between actual and predicted values as a function of prediction time where prediction time is given as discrete time steps into the future. The correlation coefficient between actual and predicted values is defined in the standard statistical way and is widely used as a measure of predictive skill. This procedure is followed for both the neural network model and for the optimum autoregressive model. Results are shown in Fig. 3a. For the first few steps into the future predictions from both models are good and the difference between the two models in terms of predictive skill is small. However, the neural network makes significantly better forecasts than does the autoregressive model as prediction time increases. The neural network model maintains greater predictive skill compared with the autoregressive model throughout the entire forecast period. The autoregressive model is essentially a linear model and therefore incapable of capturing the inherent nonlinear nature of such a signal. Since the signal is, in fact, chaotic we cannot hope to make accurate predictions with any model too far into the future.

Recently it has been suggested that certain nonlinear prediction techniques are capable of distinguishing between chaos and noise in time-series records [4]. We demonstrate this capability with neural networks by comparing the above results with results from a model trained on a time series generated from discrete points on a sine wave having a unit amplitude and adding to it at each step a uniformly distributed random variable in the interval [-0.5, 0.5]. Such an example may display dynamical character similar to chaotic systems. For example, spectrum analysis will result in spectra exhibiting peaks superimposed on a continuous background and dimensional analysis may indicate anything from a low-dimensional system (if noise is weak) to a random signal (if noise is strong).

After training the neural network on the first half of the signal composed of a sine wave plus noise we make predictions on the second half and, as was done with the Lorenz system, we compute the correlation coefficient between actual and predicted values as a function of prediction time. The nearly horizontal line in Fig. 3b is the result of this procedure. The independence of predictive skill with prediction length for this example is in sharp contrast to the rapid decrease of predictive skill for a chaotic signal (also shown in Fig. 3b). From the differences we suggest that predicting time series using neural networks is another method for differentiating additive noise from deterministic chaos. With a simple autoregressive model, predictive skill on a time series containing periodicities and/or noise will show a marked dependency on prediction length making them inappropriate for distinguishing chaos.

### III. CONCLUSIONS

In applying chaos theory in the analysis of time-series data one usually begins by estimating the dimension of the underlying attractor [10, 11, 12, 13]. This is done by constructing a state-space embedding from the time series and then applying some variant of the correlation algorithm [14]. The dimension, which is given by the power-law (scaling) behavior of the correlation integral, gives a measure of the effective number of degrees of freedom of the system. Because the scaling regions used to estimate the dimension involve only a small number of distances between points in the state space much of the information in the time series is lost which for relatively short series can cause serious problems. In addition, such methods cannot in general distinguish output from a random process from output as a result of a chaotic dynamical process [15]. In this paper we show the ability of neural networks in making time series predictions and demonstrate that such methodology is capable of distinguishing between additive noise and chaos.

Thanks are extended to Jay Perrett and John van Stekelenborg for introducing me to the back-propagation neural network and to A. A. Tsonis for stimulating discussions during the course of this research.

### REFERENCES

Spurious high frequency oscillations occur in forecasts made with the primitive equations if the initial fields of mass and wind are not in an appropriate state of balance with each other. These oscillations are due to gravity-inertia waves of unrealistically large amplitude; the primary purpose of initialization is the removal or reduction of this high frequency noise by a delicate adjustment of the analysed data. In this paper a simple method of eliminating spurious oscillations is presented. The method uses a digital filter applied to time-series of the model variables generated by short-range forward and backward integrations from the initial time.

The digital filtering technique is applied to initialize data for the HIRLAM model. The method is shown to have the three characteristics essential to any satisfactory initialization scheme: (a) high frequency noise is effectively removed from the forecast; (b) changes made to the analysed fields are acceptably small; (c) the forecast is not degraded by application of the initialization.

The digital filtering initialization (DFI) technique is compared to the standard non-linear normal mode initialization (NMI) used with the HIRLAM model. Both methods yield comparable results, though the filtering appears more effective in suppressing noise in the early forecast hours. The computation time required for initialization is about the same for DFI and NMI. The outstanding appeal of the digital filtering technique is its great simplicity in conception and application.
GRID SIZE INDEPENDENT CONVERGENCE OF THE MULTIGRID
METHOD APPLIED TO FIRST ORDER PDES.

Per Lötstedt
Saab Aircraft Division, SAAB–SCANIA, S–581 88 Linköping, Sweden,
and

Abstract.
The multigrid method for numerical solution of first order partial differential equations is analyzed. The smoothing iterations are of polynomial type and are used on all grids, also the coarsest. It is shown that the convergence is independent of the grid under certain conditions. This is a result of a propagation of smooth error waves out of the computational domain and a damping of oscillatory error modes. Numerical experiments illustrate the theoretical results.

Introduction.
When the multigrid method is used for the numerical solution of systems of first order partial differential equations such as the Euler equations of fluid flow, it is not necessary to solve the equations exactly on the coarsest grid to obtain good convergence characteristics, see e.g. [3]. For elliptic equations all proofs of grid independent convergence assume that the solution is determined exactly on the coarsest grid [2]. Here we explain why the behaviour of the multigrid method is different for the two types of equations and prove that under certain sufficient conditions the convergence rate is independent of the grid also for first order equations.

Wave propagation and damping.
The multigrid strategy adopted here is a V-cycle with a few pre- and postsmoothing steps before going to and after returning from the next coarser grid. Also on the coarsest grid a few smoothing steps are taken. Let the discretization of our partial differential equation be

\[ Q u = f. \]

For the convergence of linear problems it is sufficient to study the case \( f = 0 \), which is equivalent to looking at the convergence to 0 of the initial error. The relaxation scheme \( S \) on grid \( I \) is assumed for smooth functions to be such that

\[ \hat{S}_I u = \delta \Delta t (\hat{\omega}_I u; \hat{\omega}_I u; \hat{\omega}_I u) \]

where a hat denotes the Fourier transform and \( \Delta t \) is a small parameter like a time step. Iterative methods of this kind are Runge-Kutta time stepping [3] and the Chebyshev method [7]. For systems of partial differential equations with constant coefficients in two space dimensions it is shown in [5, 2] means of Fourier analysis that oscillatory parts of the error on the finest grid are damped efficiently but the smooth parts of the error are not damped very well. The difference between first order and elliptic pdes is that in the first case the smooth part of the error is propagated out through open boundaries. This propagation is achieved with a time step

\[ \Delta t = \sum_{i=1}^{L+1} \Delta t_i, \]

where \( L+1 \) is the number of grids. This was proved in [1] for \( L=1 \), one space dimension and a scalar equation. It is shown in [4, 5] that the behaviour of the smooth part of the error is governed by the differential equation. Let us keep the coarsest grid fixed and add finer grids by halving the step size. Then the smooth part, which is well represented on all grids, is transported out through the computational boundaries at a rate which is determined by the differential equation and the accumulated time-step. A lower bound on this rate is given by the rate on the coarsest grid. Furthermore, the damping of the oscillatory error modes can be made independent of the finest grid. In [4] an upper bound independent of the finest grid size is derived for the number of V-cycles needed for a two-dimensional scalar, constant coefficient first order pde to converge with the multigrid method. The results can be generalized to one and three space dimensions and to the W-cycle.

Numerical experiments.
A central difference approximation of the space derivative with added artificial viscosity and a Runge–Kutta time-stepping procedure as in [3] are used in the first two examples. The coarse grid with 24 points is the same in the grid sequences in the examples and the step size is doubled between a grid and the next coarser grid. There is one pre-smoothing step in the V-cycle. In fig. 1 a simple one-dimensional example is displayed. The equation is

\[ u_{xx} = 0, \]

and the initial error moves to the left through the boundary as the iterations proceed. The smooth error pulse is plotted every 5th iteration. It is damped and propagated at approximately the same rate with 1, 2, 3 and 4 grids. In [4] similar results are obtained with the Chebyshev method. The Euler equations of fluid flow are solved in a two-dimensional channel with a bump using the grids 40x12, 80x24 and 160x48 in fig. 2. The upper and lower walls are solid and the left and right boundaries are open. The Mach number of the flow from the left is 0.5. The steady state solution on the intermediate grid is shown and the rate of convergence is measured by the residual of the continuity equation (cf. [3]). No attempt has been made to optimize the performance of the method. Finally, in fig. 3 three steps of the GMRES method [6] are used as iterative smoother in the multigrid method. The equation and the discretization are the same as in fig. 1 and the solution is plotted in every step. The results with the initial bell shaped error pulse are similar to those in fig. 1 with a convergence almost independent of the grid size even if GMRES does not satisfy the conditions on the relaxation scheme with a constant time-step.
References.

Figures.

Fig. 1. The initial bell shaped error moves to the left as the multigrid iterations with Runge-Kutta time-stepping proceed (above). Eventually it disappears through the left boundary. The number of grids is from left to right, 1, 2, 3 and 4. The coarse grid is the same. The logarithm of the Euclidean norm of \( u \) as a function of the number of V-cycles is presented showing the grid independence (left).

Fig. 2. The intermediate grid and isobars of the Euler solution are displayed to the left. The convergence history for the multigrid method with Runge-Kutta time-stepping for 1, 2 and 3 grids is shown to the right and the coarse grid is the same in all cases. The logarithm of the Euclidean norm of the residual is plotted vs the number of V-cycles.

Fig. 3. The initial bell shaped error moves to the left as the multigrid iterations with GMRES proceed (above). Eventually it disappears through the left boundary. The number of grids is from left to right, 1, 2, 3 and 4. The coarse grid is the same. The logarithm of the Euclidean norm of \( u \) as a function of the number of V-cycles is presented showing the grid independence (left).
Numerical Solution of the Equations for Two Dimensional Compressible Reacting Fluid Flow

Björn Sjögren
Uppsala University
Department of Scientific Computing
Sturegatan 4B
S-752 23 Uppsala
Sweden

Abstract: We solve numerically a system of partial differential equations describing the motion of a gas in which a chemical reaction occurs. We investigate detonation wave solutions in two space dimensions. Poor resolution of the reaction layer will lead to problems due to stiffness of lower order terms. We demonstrate such a failure and suggest a remedy for it.

1. Problem to Solve

We here consider the equations

\[ \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho \varepsilon \end{pmatrix} + \begin{pmatrix} \nabla \cdot (\rho u) \\ \nabla \cdot (\rho u u) \\ \nabla \cdot (\rho u v) \\ \nabla \cdot (\rho e + p) \end{pmatrix} = 0 \]

(1)

describing a fluid in which a one step irreversible chemical reaction is taking place. \( \rho(x, y, t), u(x, y, t), v(x, y, t), \varepsilon(x, y, t) \) are the density, \( x \)- and \( y \)-velocity, and internal energy of the gas. \( \varepsilon(x, y, t) \) is fraction unreacted gas. The pressure is given by

\[ p = (\gamma - 1)(\rho - \rho_0) \]

and the temperature is defined as \( T = p/\rho \).

We restrict ourselves to two space dimensions and we have neglected heat transfer and viscous effects. We will focus on discontinuous solutions, detonation waves. For these waves the viscosity is not as important as for the slower deflagration wave solutions. The solutions computed have qualitative features agreeing with experimental data.

\( \rho_0, \gamma_1, \gamma_2, K \) are parameters, named chemical heat release, ignition temperature, \( \gamma_1 \) to \( \gamma_2 \) ratio, and equilibrium constant respectively.

2. Numerical Method

The solution to (1) is in general discontinuous. Numerical methods specifically designed for such solutions have been developed in recent years [1,2]. We use a TVD method of second order accuracy to approximate the convective fluxes [3,4], the right hand side is solved implicitly (1, 2). We use a TVD method of second order accuracy to approximate specifically designed for such solutions have been developed in recent years.

In one space dimension, we introduce the grid points \( x_j, j = \ldots, -2, -1, 1, 2, \ldots \), and the time levels \( t_n, t_{n+1}, \ldots \). The grid spacing is \( \Delta x = x_j - x_{j-1} \) and the time step is \( \Delta t = t_{n+1} - t_n \). For the equation

\[ u_i + f(u)_i = g(u) \quad -\infty < x < \infty \quad t > 0 \]

the method then takes the form

\[ u_j^{n+1} = u_j^n - \lambda \Delta x \frac{h(u_j^{n+1}) - h(u_j^n)}{\Delta t} + \Delta t \frac{1}{2} \left( g(u_j^{n+1}) + g(u_j^n) \right) \]

(2)

where \( \lambda = \Delta t / \Delta x \) and \( h(u_j^{n+1}) = h(u_j^{n+1}, u_{j+1}^{n+1}, u_{j+1}^{n+1}, u_{j-1}^{n+1}) \) is the numerical flux function, satisfying the consistency condition

\[ h(u, u, u, u) = f(u) \]

\( u^{n+1} \) is the approximation to the solution at the point \( (x_j, t_n) \). The spatial difference operator is defined as \( \Delta x u_j = u_j - u_{j-1} \). We solve the implicit equation for \( u_j^{n+1} \) using Newton's method.

The approximation in two space dimensions is similar. But for the use of a two dimensional grid \( (x_j, y_k), j = \ldots, -1, -2, 0, 1, \ldots \), \( k = \ldots, -1, -2, 0, 1, \ldots \), we may write:

\[\begin{align*}
\frac{\partial}{\partial x} (\rho u) &+ \frac{\partial}{\partial y} (\rho v) = 0 \\
\frac{\partial}{\partial x} (\rho u^2) &+ \frac{\partial}{\partial y} (\rho u v) = 0
\end{align*}\]

\( f(u) \) is the approximation to the solution at the point \( (x_j, y_k) \). The spatial difference operator is defined as \( \Delta x u_j = u_j - u_{j-1} \). We solve the implicit equation for \( u_j^{n+1} \) using Newton's method.

The approximation in two space dimensions is similar. But for the use of a two dimensional grid \( (x_j, y_k), j = \ldots, -1, -2, 0, 1, \ldots \), \( k = \ldots, -1, -2, 0, 1, \ldots \), we may write:

\[\begin{align*}
\frac{\partial}{\partial x} (\rho u) &+ \frac{\partial}{\partial y} (\rho v) = 0 \\
\frac{\partial}{\partial x} (\rho u^2) &+ \frac{\partial}{\partial y} (\rho u v) = 0
\end{align*}\]

(3)

Here we simplify, by neglecting the dependency of \( T \) on \( \rho \), (this simplification is not done in the computations.) If the temperature increases near \( T_1 \), and if \( K\Delta t \) is large, \( p^{n+1} \) will immediately drop to zero. A small change in temperature triggers a large change in \( x \) which in turn effect the temperature through the third equation in (1), the result can become a wave traveling at the unphysical speed one grid point per time step.

4. An Improved Method

A detonation wave consists of a von-neumann shock wave which first increases the temperature of the fuel mixture, so that ignition occurs behind the shock. No chemical reaction will start before the shock wave has passed through. We try to emulate this behavior in the numerical method. Numerically, there are always a few grid points in the shock. We want to make sure that none of the points inside the shock triggers the chemistry. The simplest way of ensuring this is to evaluate the right hand side a few grid points ahead of the shock. This method has been successful in one space dimension. The main point of this paper is that the method also remedies the error in our two dimensional computation above.
Figure 3 shows the same computation as figure 2, with the exception that the right hand side is evaluated one grid point out from the wave front. The direction in which to go is determined as the direction of smallest temperature. I.e., in one space dimension we evaluate

\[(p\tau)^{k+1} = (p\tau)^{k} - \lambda A k_{x}^{2} \tau_{min}^2 - \Delta t h (p\tau)^{k+1} \cdot \Delta x_{h}^2\]

with \(\lambda\) the value of \(k\) that minimizes \(\tau_{min}\) over \(k = -1,0,1\). The two-dimensional case is the straightforward generalization of this procedure.

![Figure 2: Density contour lines.](image)

We plot density contours of the solution on a domain determined by the time- and \(\gamma\)-axis. The same cellular pattern as seen on seeded plate recordings in experiments [6] appears. Figure 4, 5 and 6 show these plots for the cases discussed above. Figure 4 is computed with the small time step, figure 5 with the twice larger time step \(\Delta t = 2\Delta t_0\) and finally figure 6 is obtained using the larger time step and the modification.

![Figure 4: \(\gamma-t\) plane](image)

![Figure 5: \(\gamma-t\) plane](image)

![Figure 6: \(\gamma-t\) plane](image)

Both the run with small time step, and the modified run with larger time step predicts approximately the same size for the detonation cells.

References


A BLACKBOARD SOFTWARE DESIGN FOR CFD SOFTWARE

M. Petridis, B. Knight, D. Edwards
Centre for Numerical Modelling and Process Analysis
Thames Polytechnic
Wellington St
Woolwich, London SE18 6PF, UK

Abstract - This paper presents a software design approach for CFD software. The approach is based on the Data Structure design methodology and results in an architecture built around the Blackboard structure. An experimental prototype implementation of this approach is described with reference to the Blackboard structure as well as the integration of an Intelligent Knowledge Based System in the form of a rulebase and a set of qualitative data entries.

I. INTRODUCTION

Computational Fluid Dynamics (CFD) software is a continuously developing area in engineering software which is particularly known for its highly algorithmic nature as well as the diversity of its application fields within science and engineering. Different areas such as aerodynamics, meteorology, thermodynamics and chemical reactions modeling are among those providing a long queue of new mathematical models for solution by a CFD software package.

The need for integration of all phases and aspects of the mathematical modeling process, from problem set-up to the final results validation and interpretation as well as the quest for more reliable and accurate solutions have increased the demands on CFD software in terms of complexity and efficiency. The increasing availability and accessibility of more powerful computer hardware in terms of memory and processing speed has not managed to quench the thirst for increased power or the accuracy of the solutions. In this context, the motivation for a framework that allows and promotes the use of software modules which can be integrated and re-used in a continuous fashion ultimately leading to a software system. This architecture must be defined in terms of

(b) Integration into a software architecture. Once the database and the software modules have been properly designed and defined, there is a need for an overall architecture that can provide the operational framework of the software package. This architecture must be defined in terms of

1. The control structure
2. The modular communications structure and mechanism
3. The human interface

(c) The end user interface. The human interface must be able to respond to the user in a natural and friendly way. It should

- Communicate to the user using a set of terms and concepts in a form that is naturally perceived by him.
- Give a flexible and layered "feel" to the different users in terms of background and expertise, that may use the software.
- Integrate expert knowledge and inferencing procedures into the software.
- Give the ultimate control and decision making to the user.

One way of tackling these points is by introducing an Intelligent Knowledge Based System (IKBS) into the software. Typically IKBS systems are capable of delivering many of the requirements of such a design. They allow the integration of much of the expertise needed in such a design, in a particularly flexible and incremental way. Expertise exists in many different areas. For example in the solution method itself, in terms of speed and accuracy, and in terms of the user requirements of the software. What sort of solution does the user want? These requirements constitute what is often referred to as Intelligent Front Ends (Ristshine) to software. However it is argued here that this reference, gives the wrong view of the architecture for intelligence, since it implies i.e. it can be bolted on to existing software packages. The view we take here is that the architecture for the system must allow for a more integral IKBS component. Expertise exists at every level of CFD, from solution method to physical processing requirements, to user interface.

The integration of the software into the overall software architecture which was discussed in point (b) above, is the design decision that will affect most the resulting software quality mainly in terms of maintainability and efficiency. One such architecture which has shown considerable advantages for building CFD software packages is presented in the following sections.

The Blackboard architecture

The Blackboard architecture is a software framework that has its roots in Artificial Intelligence (AI). It is built around the Blackboard (BB) construct which is used as the main mechanism for passing information and control between software modules. As its name indicates, the blackboard is basically a mechanism that stores several pieces of information and on which a number of software modules have restricted access to write and read. As shown in figure 1, modules A and B have both access to parts of the BB and so can selectively share information they need for their function. This is a flexible way of passing data from one module to another, hiding if necessary information about the nature of these data, as each module knows where to find the information it needs. The Database Management System (DBMS) can be used as a fleet of such information providing different modules with the data they request and in the form they need them.
The Blackboard architecture for CFD software

The Blackboard architecture was implemented in an experimental prototype CFD application which was built to demonstrate this approach to CFD software design. This is a working CFD package developed to deal with a characteristic family of CFD modelling applications. It is essentially a two dimensional transient scalar multiphase finite difference code containing among others a number of numerical solvers and grid generators that can be used to tackle a class of application problems. The code is implemented in the 'C' programming language and runs under various operating systems. The database and the software modules are designed using a Data Structure oriented approach and the whole software structure is built around the Blackboard architecture as shown in Figure 2.

The information stored on the Blackboard is an abstraction of the information stored in the working database, as the Blackboard provides an abstract overview of the problem and the solution for the given modelling application together with the state of the modelling process at any given point in time as well as the dynamic of the processes and tasks that are currently executing or are waiting to be executed. As shown in Figure 2, the Blackboard system comprises of the blackboard database around which a database management system is built to allow a flexible and polymorphic view of the Blackboard data from the accessing software modules. The Blackboard itself is implemented as a hierarchical structure containing information about the problem, the solution settings and state at any point and a task buffer containing a number of tasks that have been decided to fire but do not have the opportunity to do so yet. A task is implemented as a piece of data consisting of a function and a number of data parameters or other functions that are needed for the realisation of the particular task. The operation of the Blackboard system is controlled by the main module M which looks at the solution state and the tasks buffer and if required sends a message to the appropriate module to come to the B-B system and have the particular task assigned to it. A series of modules (0-3) are working on the database and have restricted access to the Blackboard system. When a message is sent to one of them by the B-B system, that module goes to the tasks buffer and takes the next task in it together with the information needed for its execution, leaving a message to notify it that has taken up the task. After completion of the assignment the module returns to the B-B system and reports the successful or unsuccessful termination of its task.

There are different types of working modules.
- 0. Database Browsers - Pattern Recognisers that look through the amount of data in the database for patterns in the data to store in an abstract form in the B-B.
- 1. Database Constructor-Modifiers (such as grid generators) that construct the topology and instantiate the database in a dynamic fashion.
- 2. Database Executors (such as solvers) having restricted access to values of data in the database.
- 3. Output functions that collect values from the database and work on them in order to output to the user such as contouring modules.

The control module CNT asks the user for his requirements and consults the Intelligent Knowledge Based System (IKBS) as well as looks at the B-B system to have an overview of the solution state at any time. Combining the above, the control module uses its built in inference engine to decide what the next task(s) to be performed is(are), outputs the user the reasoning of this decision and puts the task(s) in the tasks buffer (B-B).

Discussion of the resulting architecture: Conclusions

This architecture has shown to possess a number of advantages. It provides a flexible way of intermodular information and control passing. It gives a natural separation and data uncoupling of the control and user interface modules from the number crunching modules which enhances flexibility and allows for easier maintenance of the software because of the hiding of important decisions mainly with respect to the data structure. It contains an abstract representation of knowledge about the problem and its solution at any point as a subset of the huge amount of raw data usually associated with CFD software. Information is in a form closer to the natural perception of the user and a way that expert rules are formulated. This makes the task of integrating an IKBS into the software easier.

An IKBS has been integrated in the experimental CFD software prototype comprising a Data Dictionary and a Rulebase which are accessed by the control module (CNT) (Figure 2) and interpreted by the Inference Engine which is built in the module.

The Data Dictionary is a collection of entities describing qualitative data formally defining and quantifying them. Concepts such as "Fine grid" or "slow convergence" which can be useful for the natural description of a state and the formulation of expert rules are defined in the Data Dictionary. The Rulebase is a collection of expert rules that are formulated as Prolog clauses and describe in a heuristic or production manner the knowledge that can be used to infer actions to be taken by the software during the various stages of the modelling process. So there are rules to decide the desirability technique of the solver to be used and to monitor and accelerate the successful modelling process changing settings and parameters. Two examples follow:

- Setting up the solver
  Is(Model,"Laplace") if Is(Time_dependency,"steady_state") and
  Is(Phases,1)
  and
  Solver("rectangular","lineSOR")

- Determining the successful conclusion of the solution procedure
  End Solver if Is(Convergence,"acceptable").

The truth of predicates in the rules is determined in a usually recursive fashion looking up entries from the Data Dictionary and comparing them to qualitative data furnished from the working database. A set of tools has been integrated to the prototype comprising of Data Dictionary and Rules editor and a compiler which translates the data entries and the rules respectively to "C" structure and "C" reactive structures so that different Rule sets and Data entry sets can be embedded in the software to deal with different areas of CFD modelling.

There is obviously a higher communications overhead associated with the use of the B-B system but due to the CPU intensive number crunching nature of CFD software this does not affect substantially the overall efficiency of the software but experience has shown that there are a lot to be gained by this approach to CFD software design although more work is needed in validating and extending the existing prototype and looking for ways to optimize its overall performance.

References

4. Knight, B., A Mathematical Basis for Entity Relationship Approach to Software Engineering ed. Davis, pp. 1:
8. Morgan, T., Englemore, R., Blackboard Systems, Addison-Wesley, 1983

597
The Role of Physical Analysis in the Numerical Simulation of Aerodynamic Flows

Hanzhin Zhang
China Aerodynamic R & D Center
P. O. Box 311
Mianyang, Sichuan 621000
PRC

Fengcan Zhuang
Chinese Aerodynamic Research Society
P. O. Box 2425
Beijing, 100089
PRC

Abstract

Typical examples are given for the illustration of the role of physical analysis in the numerical simulation of aerodynamic flows. The first is related to the proper choice of grid Reynolds number and avoidance of spurious oscillations or even chaos in a numerical calculation of shock waves with shock capturing technique. The second shows the development of a vortex along its axis, the existence of unstable limit cycle and its relation to flow picture in the bursting region.

I. Introduction

With the rapid development of supercomputers it seems that numerical results are easily obtained. The essential problem is that whether the results thus obtained are physically in reality and the large quantity of numerical data if improperly treated would lead to distorted physical flow pictures which may often be misleading. Thus the treatment given in this paper may not be without practical significance. Examples are given for the study of flows containing shock waves and vortex flows.

II. Solution of one dimensional shock wave

For steady flow in one dimension, the Navier Stoke's equations can be integrated yielding

\[ \frac{du}{dx} = \beta \frac{(u - u_{\infty})(u - u_2)}{u} \]

where

\[ \beta = \frac{3(\gamma + 1)}{\gamma - 1} \frac{p_{\infty}u_{\infty}}{\mu} \]

\[ u_{\infty} = \frac{\gamma - 1}{\gamma + 1} \left( 1 + \frac{2}{(\gamma - 1)M_{\infty}^2} \right) \]

In order to reveal the possibility of producing chaos we first use perturbation technique and keep only second order terms, we can see immediately the equation is reduced to Landau's equation. Let us suppose the initial value be given at far upstream and we are looking forward: the solution as we march downstream. Then it is possible to put

\[ u = 1 + z \]

Substituting eq.(5) into eq.(3) and keeping terms up to \( O(z^2) \) we have

\[ \frac{dz}{dx} = \beta (1 - u_2)z + \beta u_2 z^2 \]

which is indeed a Landau equation. Write down corresponding finite difference equation in the following form

\[ z^{n+1} = z^n + \beta \Delta x (1 - u_2)z^n + \beta u_2 z^n (z^n)^2 \]

Introducing a new variable \( z \)

\[ z = - \frac{u_2 \beta \Delta z}{1 + \beta (1 - u_2) \Delta z} \]

We have

\[ z^{n+1} = \alpha_n z^n (1 - z^n) \]

and

\[ \alpha_n = 1 + \beta (1 - u_2) \Delta z \]

We immediately recognize that if \( \alpha_n > 3, i.e. \)

\[ \beta \Delta z > \frac{2}{1 - u_2} = r_* \]

period doubling will occur. We note \( \beta \Delta z \) is essentially the grid Reynolds number, \( r_* \) depends on \( \gamma \) and \( M_{\infty} \), take for example \( \gamma = 1.4 \) and \( M_{\infty} = 4 \), then \( r_* = 2.56 \). The period doubling phenomenon has been verified by numerical simulation of one-dimensional Navier Stokes equation. Similar conclusion can be drawn if we start the solution at downstream side and marching toward the upstream.

III. The Development of a Vortex along its Axis

Suppose we have a vortex spiralling around its axis \( z \). We now study the flow pictures in the transversal planes, where \( x, y \) and \( z \) form a local cartesian coordinates with the origin at 0. Along \( z \) axis th- transversal velocity components \( u \) and \( v \) are zero. Let us consider the flow is steady, the equation for the streamline is

\[ \frac{dx}{u(x,y,z)} = \frac{dy}{v(x,y,z)} = \frac{dz}{w(x,y,z)} \]

The following results can be easily obtained.

1. If \( \frac{\partial w}{\partial x} \) has the same sign in the neighborhood of the origin (either always positive or always negative), then in this region the transversal section streamlines will not be closed.

2. Depending on whether \( \frac{\partial w}{\partial x} \) is positive or negative, we will have the transversal streamlines being spiral outward or inward, and in the usual cartesian we call spiral inward vortex as stable and spiral outward vortex as unstable.

3. At the point of a vortex bursting \( pw = 0 \), so there must be a decellerating flow just before bursting to occur and vortex bursting is accompanied with an unstable spiral.

4. From a similar analysis in nonlinear mechanics, we know there exists a stable limit cycle in front and in the vortex bursting region. In case we have a bursting bubble, then there must exist a second limiting cycle, through this limit cycle is unstable. So there is no guarantee for the actual realization of steady vortex bursting bubble.

5. In the longitudinal section at bursting point, the section streamlines are of saddle type.

We have made a systematic analysis of available experimental data related to vortex bursting and also made a study of flow around an axisymmetric obstacle placed on a flat plate. The numerical results agree with all qualitative pictures mentioned above.

IV Conclusions

Though the physical analysis made so far involving little mathematics yet it already shows the potential involved to understand complex flow pictures, otherwise we will be lost in an ocean of numerical data. It is also essential to uncover the underlying physical mechanism and to lead to effective means to control or modify the flow to suit our specific purposes.

The research was supported by the China National Natural Science Foundation.
Abstract—For improving the resolution of the shock, a dispersion regulator and an artificial dispersion are introduced, and an upwind compact scheme with dispersion regulator is developed.

1. INTRODUCTION

In ref. 1 a compact scheme was developed. This scheme was useful for improving both of the accuracy and the efficiency, but there are oscillations in the numerical solutions. In this paper a dispersion regulator and artificial dispersion are introduced and an upwind compact scheme with dispersion regulator is developed. Numerical experiments show the scheme developed has high resolution.

2. UPWIND COMPACT SCHEME WITH DISPERSION REGULATOR

A. Compact Scheme and Upwind Compact Scheme

Consider the model equation

\[ \frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = 0, \quad f = au, \quad a = \text{const.} \]

In ref. 1 a compact scheme is developed. The scheme has high accuracy, but there are oscillations in the numerical solutions near the shock. The solutions may be improved if the scheme is upwind biased. Consider a semi-discrete approximation for equation (1)

\[ \frac{\partial u}{\partial t} + \frac{1}{2} \frac{1}{\Delta x} F_j = 0 \]

\[ \omega_0 F_{j+1} + \rho_0 F_j + \rho_0 F_{j-1} = 0, \quad \omega_0 = \frac{\Delta x}{2} - 2 \xi_0 \]

\[ \omega_0 = \frac{\Delta x}{2} - 2 \xi_0, \quad \xi_0 = \frac{1}{6} - 2 \xi_0 \]

\[ \rho_0 = \frac{\Delta x}{2} + 2 \xi_0 \]

where \( \xi \), \( \xi^- \) and \( \xi^+ \) are forward, backward and central differences. Scheme (2) and (3) for \( a > 0 \) with \( \xi > 0 \) and \( \xi^- > 0 \) is called upwind compact scheme. With different choice of the parameters \( \xi \) different schemes can be obtained. The simplest one within schemes with accuracy of order three is

\[ \frac{\partial u}{\partial t} + \frac{1}{2} \frac{1}{\Delta x} F_j = 0 \]

It is dissipative and it can be solved easily.

B. Upwind Compact Scheme with Dispersion Regulator

Consider a semi-discrete second order accurate scheme

\[ \frac{\partial u}{\partial t} + \frac{1}{2} \frac{1}{\Delta x} F_j = 0 \]

The modified equation with the leading term takes form

\[ \frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = \Delta x \frac{\partial^2 u}{\partial x^2} (R \frac{\partial^2 u}{\partial x^2}) \]

Rewrite it into form

\[ \frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = \Delta x \frac{\partial^2 u}{\partial x^2} (R \frac{\partial^2 u}{\partial x^2}) \]

If we have a shock as shown in Fig. 1 the coefficient \( M > 0 \) in the front of the shock. Negative \( M \) is a reason of oscillation production behind the shock in the numerical solutions. In the present paper the parameter \( R \) in (6) is reconstructed as

\[ R(u) = \frac{\left| u(x + \Delta x) - u(x) \right| - \left| u(x) - u(x - \Delta x) \right|}{\left| u(x + \Delta x) - u(x) \right| - \left| u(x - \Delta x) - u(x) \right|} \]

which is called dispersion regulator in this paper. The function \( R(u) \) changes sign across the shock.

With dispersion regulator we have two kinds of schemes:

a. Scheme with 'artificial dispersion. The scheme suggested is

\[ \frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = \Delta x \frac{\partial^2 u}{\partial x^2} (R \frac{\partial^2 u}{\partial x^2}) \]

b. Upwind compact scheme with dispersion regulator. The parameters \( \xi_k \) are determined as follows

\[ \xi_2 = 0, \quad \xi_0 = \xi_0 R(u), \quad \xi_0 = 0.125 \]

where \( \xi_0 \) is a positive parameter and \( R(u) \) is determined by equation (7).

3. APPROXIMATION OF THE EULER EQUATIONS

Consider the 1-D Euler equations

\[ \frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} = 0 \]

\[ U = (\varphi, \varphi u, \varphi, \varphi, \varphi + p, \varphi, \varphi + p, \varphi + p) \]

\[ p = \frac{1}{\gamma M_0^2} \varphi, \quad E = \varphi (C_v T + \frac{\varphi}{\gamma}) \]

The difference approximation used is

\[ \left(I + \frac{\beta}{\Delta x} \frac{\partial^2}{\partial x^2} A^- \right) \frac{\partial U}{\partial j} = \frac{\partial F}{\partial j} \]

\[ \frac{\partial F}{\partial j} = \frac{\partial^2}{\partial x^2} \left( \frac{\partial}{\partial x} \right)^2 \]

This work supported in part by the National Foundation of Natural Science
where $\Lambda$ is the Jacobian matrix and $f^2 = \Lambda^T U$, $\Lambda^+ = S^{-1} \Lambda^T S$, and $\Lambda^2$ are the diagonal matrices with elements

$$\lambda_k^+ = (\lambda_k^+ \lambda_k^-) / 2$$

$\lambda_k$ are the eigenvalues of the matrix $\Lambda$. In the computation the parameters $\varepsilon_k^\pm$ are determined as

$$\varepsilon_2^- = 0, \quad \varepsilon_3^\pm = \varepsilon_0 R(\varepsilon), \quad \varepsilon_4^\pm = \pm 0.125$$

4. NUMERICAL EXPERIMENTS

A. 1-D Shock Tube Problem

The upwind compact scheme was used to solve this problem. The computed results at the time $t=0.14$ are given in Fig.2.

B. 2-D Shock Reflection

The free Mach number is $M_\infty = 2.9$, and the incident angle is 29 degree. Some computed results are given in Fig.3. The numerical experiments show

a. the third order accurate scheme (2) and (4) has nice solutions near the shock but still with small oscillations;

b. the third order accurate upwind compact scheme with dispersion regulator has high resolution of the shocks.

C. Supersonic Flow Around Blund Body

The scheme developed in ref.2 with artificial viscosity and with artificial dispersion was used to solve the axial symmetrical supersonic flow around a sphere-cone. The shock near the axis oscillates in the numerical solution obtained with artificial viscosity. When the scheme with artificial dispersion is used non oscillations are found in the numerical solutions.

REFERENCE

1. Ma Yanwen and Fu Dexun, AIAA paper No.87-1123.
NUMERICAL SIMULATION OF SUPersonic SEPARATED FLOW OVER BLUNT CONE AT HIGH ANGLE OF ATTACK

Shen Qing, Gao Shuchun, and Zhang Hanxin
China Aerodynamics Research and Development Center
P.O. Box 211, Mianyang, Sichuan 621000, P. R. China

Abstract—Through a study for the properties of boundary layer equations, a space-marching technique for solving parabolized Navier-Stokes (PNS) equations has been developed by Zhang recently, and the axisymmetric flows over blunt cones have been simulated successfully. In the present paper, the above technique is extended to the calculation of flow over blunt cones with cross-flow separation at high angle of attack.

Introduction
Flow fields with cross-flow separations which dominate the shock layer during the reentry of configurations at high angles of attack can be simulated numerically by solving Navier-Stokes (NS) equations, where Euler cases become invalid. Because the cost is very high and the computer storage is limited, the application of time dependent NS solver to the three-dimensional configurations is restricted greatly. Thus, discussion about various simplification for the NS equations has been made by several researchers. This analysis shows that the PNS equations are of the most interest. The reasons for this are (1) falling between the complete NS equations and the boundary layer equations, the set of equations are applicable to both inviscid and viscous flows, which means that the incompressible viscous interaction is included automatically; (2) the PNS equations are hyperbolic parabolic type when certain conditions are met; (3) the inviscid region of the flow is supersonic, and the streamwise velocity component is positive, and then can-be solved numerically during the reentry of configurations at high angles of attack.

The conservative form of the PNS equations is

\[ \rho \frac{\partial u}{\partial x} + \frac{\partial p}{\partial x} = 0 \]

where the perfect gas assumption is taken, \( \gamma = 1.4, \ P_T = 0.72, \ T_T = \gamma M_\infty^2 P_\infty / \rho \) and the coefficient of viscosity (\( \mu \)) is determined from Sutherland's law. Numerical Algorithm

Eq. (1) can be rewritten as

\[ B^* + P_4 + \frac{g_4}{4} = R_3 S_4 \]

where \( B^* = E - P \), and

\[
P = \begin{bmatrix}
0 \\
(1-\omega) \xi_0 \\
0 \\
0
\end{bmatrix}
\]

\( \omega = \text{min} (1, \kappa M_\infty^2) \), \( \kappa \) is a safety factor and is taken as 0.8 in the calculation. For computing efficiency, an Euler implicit approximately factored, finite difference algorithm in delta form is used, where \( P_4 \) is treated explicitly. The difference equations of Eq. (2) are

\[
\Delta x \left[ \frac{C_1^+ + \Delta \xi (\Delta b_1^+) \Delta t}{1 - \omega} \right] = - \left( C_1^+ + \Delta \xi (\Delta b_2^+) \right) \Delta p + \Delta x \left[ \frac{C_1^+ + \Delta \xi (\Delta R_3^+ \Delta b_1^+) \Delta t}{1 - \omega} \right]
\]

According to Zhang's technique, the pressure is calculated from Eqs. (3) by the fourth-order dissipation term of [3] is employed to suppress high frequency oscillations. Eqs. (3) represent block tri-diagonal systems of equations which are solved using a matrix solver as described in [13]. It is important that the main diagonal elements occupy a dominant position in solving the equation. Usually, implicit smoothing terms are used to enhance the main diagonal element. The same purpose is attained with limiting the marching step size, then no implicit smoothing terms are needed and higher accuracy is achieved. Also, the implicit NND scheme has been used, and no obvious difference is found between results given by two methods separately.

Boundary Conditions
The body surface conditions (wall slip, a specified surface temperature, and zero normal pressure gradient) are implemented implicitly in the algorithm.

The outflow shock is fitted using the "pressure approach" procedure. The pressure behind the shock is calculated from Eqs. (3) with one-side difference in the z-direction, and the remaining flow variables are evaluated by satisfying the Rankine-Hugoniot relations.

As symmetry-plane, the reflected condition is employed implicitly.

Initial Conditions
Initial conditions for starting downstream marching computation are provided with the three-dimensional time-dependent NS code.

Results and Discussion

Case 1
At first, the flow over a blunt sphere cone with 10° cone half-angle at no incidence is calculated in order to demonstrate the efficiency of the algorithm. The flow conditions are \( M_\infty = 8, \ R_e = 5 \times 10^4 (L/R_e = 50) \), \( T_w = 72.46 K, \ T_a = 200 K \). The results show good agreement with the data from [14] for the shock location in Fig. 1 and surface pressure in Fig. 2, which verifies the validity of the present code.

Case 2
Flow over a blunt sphere cone with 4° cone half-angle for an angle of attack of 20° is computed here. Flow conditions are \( M_\infty = 10, \ R_e = 2.3 \times 10^4 (L/R_e = 10.52), \ T_w = 49 K, \ T_a = 29 K \). Results are compared with those of [15]. Fig. 3a presents the shock shape and the pressure distribution on the pitch plane. Fig. 3b is an experimental photograph. Fig. 4 and 5 present the surface pressure and heat transfer rate respectively. The comparison of calculated and measured surface pressure variations with the circumferential angle at different axial positions is presented in Fig. 2, and the similar comparison for the normalized heat-transfer rate is shown in Fig. 7. The comparison shows the good agreement between the computational and experimental results.

No adverse pressure gradient along the meridional angle exists at x/L = 0.45. However, a minimum pressure point develops off the lee meridian at x/L = 0.45, which indicates the existence of the crossflow separation. As the result of the appearance of the secondary crossflow separation, the second minimum pressure point occurs between the first minimum point and the lee meridian at x/L = 0.55. Fig. 8a presents the limiting streamline where the development of the open separation is observed. The primary group of converged limiting streamlines and the second group of converged limiting streamlines manifest the primary and secondary separation lines. The experimental photograph (Fig. 8b) of [15] which provides us the real off-flow pattern shows the same image. The vortex structure is given by plotting crossflow velocity vectors in Fig. 9.
Conclusion

The space-marching technique developed by Zhang for the axisymmetric flows over blunt cones is extended to simulate asymmetrical flows with crossflow separation. The results for blunt cones, especially the separated flow patterns calculated, are obtained satisfactorily. It is shown that the PNS method used in this paper is successful.

References

A FAST EFFICIENT ALGORITHM
FOR THREE DIMENSIONAL HYPERSONIC VISCOUS FLOW

RU-QUAN WANG
Computing Center
Academia Sinica
Beijing, 100080, CHINA

JU-KUI XUE
Department of Physics
Northwestern Normal University
Lanzhou, 730070, CHINA

Abstract Based on the Single Level Conservative Supra-characteristic method (CSCM-S) proposed by Lombard et al. [1], we suggest a more fast efficient algorithm, which combines a single-marching technique for super-sonic dominant zones with a multi-sweep procedure for complex flowfields. The new one requires about an order of magnitude less CPU time than the CSCM-S algorithm.

I. INTRODUCTION

Over the past ten years there has been a great development in efficient algorithms for compressible flow. Among them, the CSCM-S algorithm [1] is very attractive for multidimensional Euler and Navier-Stokes problems. The algorithm is based on an implicit symmetric Gauss-Seidel method along the very attractive for multidimensional Euler and Navier-Stokes problems. The algorithms for comprissible flow. Among these the CSCM-S algorithm is chosen as the basic scheme

\[
\begin{align*}
\Delta E &= \Delta E + \Delta E^* = \hat{A}^*V_{ij} + \hat{A}^* \Delta q_i \\
\Delta F &= \Delta F + \Delta F^* = \hat{B}^*V_{ij} + \hat{B}^* \Delta q_i \\
\Delta G &= \Delta G + \Delta G^* = \hat{C}^*V_{ij} + \hat{C}^* \Delta q_i 
\end{align*}
\]

where \( V_{ij} \) and \( \Delta q_i \) are the forward and backward differences, \( A = \partial F/\partial q_j \), \( B = \partial E/\partial q_i \), \( C = \partial G/\partial q_j \). The CSCM flux-difference splitting [1] is chosen as the basic scheme of our algorithm, i.e. the inviscid flux difference can be expressed as

\[
\begin{align*}
\Delta E &= \Delta E + \Delta E^* = \hat{A}^*V_{ij} + \hat{A}^* \Delta q_i \\
\Delta F &= \Delta F + \Delta F^* = \hat{B}^*V_{ij} + \hat{B}^* \Delta q_i \\
\Delta G &= \Delta G + \Delta G^* = \hat{C}^*V_{ij} + \hat{C}^* \Delta q_i 
\end{align*}
\]

II. SIMPLIFIED NAVIER-STOKES EQUATIONS

We consider an explicit-implicit difference equation for

\[
J = \partial q_i/\partial t + \partial E/\partial x + \partial F/\partial y + \partial G/\partial z = \frac{1}{Re} \partial F/\partial \eta + \partial G/\partial \delta
\]

where \( J = (p, \rho, \rho v, \rho w, \rho e) \), \( e = \frac{1}{2} \rho v^2, T = \frac{1}{p} (e + \frac{p}{\gamma - 1}) \), Re = \( \frac{\mu}{\rho v} \), and the new developed scheme contains such viscous terms that have orders of magnitude (4).

III. NUMERICAL METHOD

A. MULTIPLE SWEEP PROCEDURE

We consider an explicit-implicit difference equation for (1) at any grid point \( (\xi_i, \eta_j, \zeta_k) \)

\[
\frac{\partial \xi^{i+1}}{\partial t} + \frac{\partial E^{i+1}}{\partial x} + \frac{\partial F^{i+1}}{\partial y} + \frac{\partial G^{i+1}}{\partial z} = \frac{1}{Re} \frac{\partial F^i}{\partial \eta} + \frac{\partial G^i}{\partial \delta}
\]

The CSCM flux-difference splitting [1] is chosen as the basic scheme of our algorithm, i.e. the inviscid flux difference can be expressed as

where \( V_{ij} \) and \( \Delta q_i \) are the forward and backward differences, \( A = \partial F/\partial q_j \), \( B = \partial E/\partial q_i \), \( C = \partial G/\partial q_j \). Jacobian matrices and the matrices \( \hat{A}^*, \hat{B}^*, \hat{C}^* \)

\[
\begin{align*}
\hat{A}^* &= \hat{A}^*V_{ij} + \hat{A}^* \Delta q_i \\
\hat{B}^* &= \hat{B}^*V_{ij} + \hat{B}^* \Delta q_i \\
\hat{C}^* &= \hat{C}^*V_{ij} + \hat{C}^* \Delta q_i 
\end{align*}
\]

Note that the forward-sweep step (6) may be removed for the super-sonic dominant flowfield and the numerical solution is found using only the forward-sweep (5). In fact, this can be easily done putting \( \hat{A}^* \equiv 0 \) in eq.(4) and numerical results in this way are in good agreement with those obtained by the PNS space-marching technique. In this case the forward sweep step becomes the following form

\[
\begin{align*}
&(-\hat{B}', \hat{D}, \hat{B}') \delta q^{(n+1)}_i = -L^{(n+1)} q \\
&\delta q^{(n+1)}_i = q^* + \delta q^{(n+4)}_i, \quad \delta q^{(n+1)}_i = 0
\end{align*}
\]

where

\[
L^{(n+1)} q = -L_0 q + \hat{A}^* (\hat{q}^{(n+1)}_i - \hat{q}^{(n-1)}_i) + \hat{A}^- (\hat{q}_i^{(n+1)} - \hat{q}_i^{(n-1)})
\]

D. SINGLE SWEEP TECHNIQUE

Note that the backward-sweep step (6) may be removed for the super-sonic dominant flowfield and the numerical solution is found using only the forward-sweep (5). In fact, this can be easily done putting \( \hat{A}^* \equiv 0 \) in eq.(4) and numerical results in this way are in good agreement with those obtained by the PNS space-marching technique. In this case the forward sweep step becomes the following form

\[
\begin{align*}
&(-\hat{B}', \hat{D}, \hat{B}') \delta q^{(n+1)}_i = -L^{(n+1)} q \\
&\delta q^{(n+1)}_i = q^* + \delta q^{(n+4)}_i
\end{align*}
\]

\[
L^{(n+1)} q = -L_0 q + \hat{A}^* (\hat{q}^{(n+1)}_i - \hat{q}^{(n-1)}_i)
\]

This work is supported by the National Natural Science Foundation of China.
In order to test the efficiency of the CSCM-C algorithm, we computed some large-scale two and three dimensional hypersonic flows past a sphere-cone and a simple space shuttle orbiter. A part of results for the sphere-cone is given here. In the paper [3] numerical test was accomplished for the sphere-cone hypersonic flow at $M_{\infty}=20$ and zero angle of attack with the CSCM-S algorithm and the convergent solution was reached through 200 global sweeps. The same problem has been also computed by the CSCM-C under the following freestream conditions:

$M_{\infty}=70$, $Re=0.6 \times 10^6$, $T_{\infty}=256k$, $T_m=1000k$

Numerical solution was found by using the single sweep technique except for a small nose region, in which the steady solution was obtained through 200 global sweeps. Figs 1 and 2 show surface pressure and heating-rate distributions, respectively. They are in excellent agreement with those obtained by our 3-D VS code [4]. The similar test was carried out for $15^\circ$ sphere-cone three-dimensional flow at $20^\circ$ angle of attack and the freestream conditions are the same as [5], i.e.

$M_{\infty}=10.6$, $Re=1.318 \times 10^6$, $T_{\infty}=47.34k$, $T_m=300k$

In this case the explicit-implicit scheme may reduce one-third of CPU time in comparison with the full implicit scheme (see table).

**TABLE COMPARISON OF DIFFERENT SCHEMES AT ONE MARCHING STATION**

<table>
<thead>
<tr>
<th>SCHEME</th>
<th>EXPLICIT-IMPLICIT</th>
<th>IMPLICIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU of per time step (sec.)</td>
<td>10.36</td>
<td>24.93</td>
</tr>
<tr>
<td>CPU of per grid point</td>
<td>0.021</td>
<td>0.032</td>
</tr>
<tr>
<td>total number of iteration</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>total CPU time</td>
<td>163.32</td>
<td>299.16</td>
</tr>
</tbody>
</table>

Figs. 3 and 4 represent axial distribution of surface pressures and heating rates along different meridianal planes. The points indicate experimental data. Clearly, the single-sweep technique is acceptable in the supersonic dominant flowfields.

**V. CONCLUSION**

As it is well known that the time-dependent Navier-Stokes solver requires a large amount of CPU time to obtain a convergent numerical solution and it is more adequate to local complex flowfields. The space marching PNS algorithm can save an order of magnitude CPU time in comparison with the complete NS solver. However it has the inherent problem with stability, i.e. the exponentially growing solutions will occur if the streamwise pressure gradient is retained in the subsonic regions. The present CSCM-C algorithm consists of the space-marching technique and the time iteration and it is free of instability and requires less CPU time than the time-dependent Navier-Stokes solver. We expect the present algorithm would be more efficient to large-scale flowfields around the missile space shuttle orbiter configurations.

**VI. REFERENCES**

NUMERICAL SIMULATION FOR 2-D FLUIDS USING NND SCHEMES

Ye Youde, Guo Shiqian and Zhang Hanxin
China Aerodynamics Research and Development Center
P. O. Box 211, Mianyang, Sichang, 621000 P. R. China

Abstract: NND scheme which is developed by Zhang has been applied to the hypersonic flow around a shuttle-orbiter-like geometry proposed by CARDC. Euler equations in a finite difference are solved by using explicit-scheme-marching algorithm and implicit marching iteration algorithm. Numerical calculations are performed for the problem under the conditions that Mach number is 7.0 and the angle of attack is 0 degree. Numerical results of these two algorithms are compared with each other and agree very well.

Introduction
In the calculation of complex flow fields containing shock waves, no less attention has been paid to the shock capturing method. In order to capture shock waves smoothly without spurious oscillations near or in the shock region, mixed dissipative schemes including free parameters with first order accuracy near shock and second order schemes elsewhere have been widely used. There are inherent disadvantages in employing these schemes. Firstly, the free parameters are basically determined by numerical experiments. Secondly, the resolution of shock is not very satisfactory. Naturally, the development of non-oscillatory dissipative schemes containing no free parameters with high resolution has much been emphasized recently, such as TVD schemes (9-11) and ENO schemes (12). Through a study of the one dimensional Navier-Stokes equations, it was found that the spurious oscillations occurring near shock waves with finite difference equations are related to the dispersion term in the corresponding modified differential equations. If the sign of the dispersion coefficient is properly adjusted in order to make the sign change across the shock waves, the undesirable oscillations can be totally suppressed. Based on this discovery, the non-oscillatory and dissipative scheme containing no free parameters (NND scheme) is developed. This is one of "TVD" schemes. In order to test the effectiveness of the scheme in space-marching calculations, we have carried out numerical simulations for two dimensional flows of shock and expansion wave interaction flowfields. Moreover, the NND schemes have been employed for the calculation of hypersonic inviscid flow around the shuttle-orbiter-like geometry.

Numerical Algorithm
1. Semi-discretized NND scheme
For an one dimensional Euler equations,
\[ \frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = 0 \]  
here U is vector and F is a function of the vector U. The NND scheme of semi-discretized form is (10),
\[ \frac{\partial U}{\partial t} = -\frac{1}{\Delta x} \left( \frac{F_{n+1}}{n+1} - \frac{F_n}{n} \right) \]  
where
\[ F_{n+1} = F_n + \frac{\Delta t}{2\Delta x} (\Delta F_{n+1} + \Delta F_{n}) \]
\[ F_{n+1} = F_n - \frac{\Delta t}{2\Delta x} (\Delta F_{n+1} - \Delta F_{n+2}) \]
\[ \Delta F_{n+1} = \frac{\Delta t}{2\Delta x} (\Delta F_{n+1} - \Delta F_{n+2}) \]
\[ \Delta F_{n+1} = \frac{\Delta t}{2\Delta x} (\Delta F_{n+1} + \Delta F_{n}) \]

2. Explicit NND scheme
For equation (1), two step predictor-corrector scheme is
\[ \frac{\partial U}{\partial t} = U_t - \frac{\Delta t}{\Delta x} (F_{n+1} - F_n) \]
\[ U_t = \frac{1}{2} (U_t + U_{t+1}) - \frac{\Delta t}{\Delta x} (F_{n+1} + F_{n+2}) \]
The scheme is second order in time and space, and is TVD scheme. The maximum allowable Courant number is 1. For simplicity, the predictor step of (3) can be also taken as
\[ \frac{\partial U}{\partial t} = U_t - \frac{\Delta t}{\Delta x} (\Delta F_{n+1} + \Delta F_{n}) \]
i.e. \( \text{minmod}(a,b) = 0 \) in (3) is used.

3. Implicit NND scheme
For one dimensional Euler equation (1), we have
\[ \frac{\partial U}{\partial t} = U_t - \frac{\Delta t}{\Delta x} (F_{n+1} - F_{n+2}) \]
\[ U_t = U - \frac{\Delta t}{2\Delta x} (\Delta F_{n+1} - \Delta F_{n+2}) \]
From (2), according to Crank-Nicolson method, \( \theta \) can be expressed as
\[ \theta = \frac{1}{2} \]

\[ \Delta_t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

This expression is second order accuracy in time for \( \theta = 1/2 \) and is first order for \( \theta = 1 \). Substituting (4) and (5) into this expression, the following implicit scheme can be obtained
\[ U_{t+1} = U_t - \frac{\Delta t}{\Delta x} (F_{n+1} - F_{n+2}) \]
\[ \theta = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}{2} \]

\[ \Delta t = \frac{1}
It is obvious that the flow is supersonic in the afterbody flowfield when the angle of attack is not very large, so the space-marching method is applied to solving the steady Euler equations. Through the transformation of coordinate system, we obtain the three-dimensional compressible steady Euler equations in dimensionless as:

\[
\frac{\partial \mathbf{u}}{\partial x} + \frac{\partial \mathbf{u} \cdot \mathbf{u}}{\partial y} + \frac{\partial \mathbf{u} \cdot \mathbf{u}}{\partial z} = 0
\]

where \(\mathbf{u}\) the coordinates of streamwise, wall-normal and circumferential directions respectively. Since equations (10) are hyperbolic and \(F = F(E), G = G(E)\) so the two-step NND scheme (3) can be used directly.

When the angle of attack is large, the subsonic pockets will arise in the afterbody flowfield, so the time and space-marching methods should be applied to solving the steady Euler equations. On the one hand, space-marching calculation is performed along streamwise direction, on the other hand, the time iteration is carried out on every cross section. It is easy to obtain implicit marching-iteration scheme from model scheme (9), the Gauss-Seidel iteration will be used in the circumferential direction.

On the body surface, all variables can be determined by using tangential flow condition and four characteristic relations derived from Euler equations. At the shock wave, the Riemann-Hugoniot and one characteristic relation can be used to determine the shock wave shape and all variables. The initial profile can be given by time-dependent blunt body code [10]. The inviscid solution is then obtained under the following initial and boundary conditions:

\[u = 7, a = 5\]

Fig. 3 gives the pressure contours on the body surface, the density contours are showed in Fig. 4. Interaction of the body shock and wing shock are showed clearly on the meridional surface \(\phi = 90, 270\). Fig. 3 and Fig. 4 also show the contact discontinuities after the interaction of body shock and wing shock. The pressure distributions on meridional surface \(\phi = 90\) obtained with explicit and implicit schemes are compared in Fig. 5, which agree very well. The above results demonstate that the capability of capturing shock and other contact discontinuities with NND schemes is satisfactory.

Concluding Remarks

According to the calculated results and the others [11-13], we are sure it is reliable to the NND scheme. The distinguished feature of the schemes is capability of capturing shock and other contact discontinuities. Furthermore, the schemes possess good characteristic and convergent accuracy, which is essential to any high shock resolution scheme. In addition, the present form seems to be the simplest, meanwhile, the amount of the numerical work is much reduced in comparison with some other high resolution TVD schemes.

Reference

[1] Pullian, T.H., AIAA 85-0438
Abstract—A dual porosity model is derived by the technique of homogenization. The model properly incorporates gravity in that it respects the equilibrium states of the medium.

1. INTRODUCTION

We consider flow in a naturally fractured reservoir which we idealize as a periodic medium as shown in Fig. 1. There are three distinct scales in this system, the pore scale, the scale of the average distance between fractures, and the scale of the entire reservoir. The concept of dual porosity [4, 6] is used to average the two finer scales in such a way that the pore scale is recognized as being much smaller than the fracture spacing scale. The fracture system is modeled as a porous structure distinct from the porous structure of the rock (the matrix) itself.

Dual-porosity models can be derived by the technique of homogenization [2, 3, 6] (see also the general references [6, 7, and 9]). Briefly, we pose the correct microscopic equations of the flow in the reservoir and then let the block size shrink to zero. The resulting macroscopic model is formulated in six space dimensions, three of them represent the entire reservoir over which the fracture system flow occurs. At each point of the reservoir, there exists a three dimensional, "infinitely small" matrix block (surrounded by fractures) in which matrix flow occurs.

For single-phase, single component flow, it is recognized that diffusive, gravitational, and viscous forces affect the movement of fluids between the matrix and fracture systems; however, only diffusive forces are easily handled (see, e.g., [1, 4, 6, 8, 9], and the many multiphase models in the petroleum literature). Simply including gravity in the matrix of the standard model [2, 3, 6] creates an inconsistency in that when the fracture system is in gravitational equilibrium, the matrix system is not.

In this paper we derive a consistent model.

2. THE MICROSCOPIC AND MACROSCOPIC MODELS

Denote the reservoir by $\Omega$. For a sequence of $\varepsilon$'s decreasing to zero, we consider equivalent representative matrix blocks that are $\varepsilon$ times the original size in any linear direction. Let $\Omega_f' \in \Omega$ be the fracture and matrix parts of $\Omega$, respectively. Each period of the reservoir is congruent to the unit cell $\varepsilon \Omega$, the period at point $x$ is denoted by $\xi(x)$. For the unit cell block $\varepsilon \Omega$ (see $\Omega_f$, $\Omega_m$) we write $\xi_f$ and $\xi_m$ for the fracture and matrix parts, respectively. Let the centroid of $\xi(x)$ be $\xi(x)$. Then $x = \xi(x) + \varepsilon y \sim x + \varepsilon y$. Asymptotically, $x \sim \xi(x)$ selects a period and $y$ specifies a point in the enlarged, congruent period $\xi \Omega$. Let $\nu$ denote the unit normal vector to the matrix fracture interface $\partial \Omega_m$ (or $\partial \Omega_f$). We use upper and lower case letters for fracture and matrix quantities, respectively. Let $\bar{p}$ (or $p$) be the fluid pressure, and $\bar{\phi}$ (or $\phi$) and $\bar{K}$ (or $K$) be the porosity and permeability on the pore scale (so $\bar{\phi} \approx 1$ and $\bar{K}$ is very large). The fracture system porosity and permeability, $\phi$ and $K$, are defined on the fracture spacing scale. Easily

$$\phi = |Q_f| \bar{\phi} / |Q|,$$

(1)

where $|\cdot|$ denotes the volume of the set, while $K$ is derived by homogenization. Finally, $(I)'$ (or $p(I)'$) and $\mu$ are fluid density and viscosity, and $y$ is the gravitational constant. Let $e_j$ point in the $j$th Cartesian direction, where $e_3$ points down.

Define the function $\psi(x)$ as the solution to

$$\nu \psi = \rho \phi g; \text{ i.e., } \int_{\Omega_f} \frac{d\nu}{\rho \phi g} = g(x_3 - x_3, 0).$$

Then $\nabla p - (\rho \phi g) \xi_3 = 0$ if and only if $p = \psi(x_3 + \varepsilon z)$ for some constant $\varepsilon z_3$, and so $\psi(x_3 + \varepsilon z_3)$ is the gravitational equilibrium pressure distribution. We note that the pseudopotential of the flow is given by $\psi^{-1}(p) - z_3$.

We ignore boundary conditions on $\partial \Omega_f$, external sources/sinks, and initial conditions since we are interested in internal flow.

The microscopic model: For the fracture flow,

$$\phi \partial p/\partial t = - \mu \varepsilon \nabla\cdot (p \varepsilon \nabla p) \left[ p - (\rho \phi g) \xi_3 \right] = 0, \quad x \in \Omega_f'$$

(3a)

$$\phi \partial p / \partial t - \varepsilon \nu \cdot \left[ \mu^{-1} \rho \varepsilon k \varepsilon \nabla p - \rho \phi g \xi_3 \right] \nabla p = \epsilon \mu^{-1} \nabla (p \varepsilon \nabla p - \rho \phi g) \xi_3, \quad x \in \partial \Omega_m''.$$ (3b)

For the matrix,

$$\phi \partial p / \partial t - \varepsilon \nabla \cdot \left[ \mu^{-1} \rho p \varepsilon k (\varepsilon \nabla p') - \rho \phi g \xi_3 \right] = 0, \quad x \in \Omega_m'$$

(4a)

$$p' = \psi^{-1}(p') + (\varepsilon - 1) (x_3 - \xi(x)) + \xi_3', \quad x \in \partial \Omega_m''.$$ (4b)

On each $\xi \Omega'(\varepsilon)$, we need to define $\xi'$. For a given $p'$, we can find for each constant $\xi'$ the solution $p'$ of the steady state problem corresponding to (4). So, for the given fracture pressure $p'$, we take the $\xi'$ which gives rise to the $p'$ that satisfies

$$\int_{\xi_m''(\varepsilon)} \phi p' \xi'' \, dx = \int_{\xi_m''(\varepsilon)} \phi p' \xi'' \, dx,$$

(5)

where $p'$ is the steady state solution of the unscaled problem corresponding to (1), given by removing the two $\varepsilon$'s appearing as coefficients in (4a) and replacing (4b) by $p' = p$. (In the case of an incompressible fluid, simply take $\xi' = 0$.)

This $\varepsilon$ family of microscopic models satisfies the following.

(i) Darcy flow governs the reservoir, and it does so in the standard way when $\varepsilon = 1$ (since then $\xi' = 0$);

(ii) For each $\varepsilon$, Darcy flow occurs in the fractures and within the sealed matrix blocks (i.e., if any matrix block $\xi_m''$ is expanded to unit size $\xi_m$, the transformed equations

---

*This work was supported in part by the National Science Foundation and the State of Texas.
indicate that Darcy flow results);
(ii) If the fracture system is in gravitational equilibrium in the vicinity of a block, then the boundary conditions on that block reflect this gravitational equilibrium;
(iii) For fixed fracture conditions around any matrix block, the steady state matrix solution gives rise to the same mass as calculated from the steady-state solution of the unscaled matrix problem.

We require (iv) so that mass is conserved, since when we scale the matrix problem with (ii)-(iii), we change the pressures which may change the total mass. Under steady-state conditions it is easy to account for any such spurious changes.

We remark that the standard microscopic model \[2, 3, 6\] replaces (4b) with \(p' = P^s\), omits (5), and to be consistent needs to have \(p'(x)g\) replaced by \(\varepsilon p'(x)g\) in (3b) and (4a). The novel expression (4b)-can be viewed as a scaled continuity of \(\psi\) needs to have \(p'(x)g\) replaced by \(\varepsilon p'(x)g\) for some \(\varepsilon\) scaling, which may change the total mass. Under steady-state conditions it is easy to account for any such spurious changes.

We remark that the standard microscopic model \[2, 3, 6\] replaces (4b) with \(p' = P^s\), omits (5), and to be consistent needs to have \(p'(x)g\) replaced by \(\varepsilon p'(x)g\) in (3b) and (4a). The novel expression (4b)-can be viewed as a scaled continuity of \(\psi\) needs to have \(p'(x)g\) replaced by \(\varepsilon p'(x)g\) for some \(\varepsilon\) scaling, which may change the total mass. Under steady-state conditions it is easy to account for any such spurious changes.

We remark that the standard microscopic model \[2, 3, 6\] replaces (4b) with \(p' = P^s\), omits (5), and to be consistent needs to have \(p'(x)g\) replaced by \(\varepsilon p'(x)g\) in (3b) and (4a). The novel expression (4b)-can be viewed as a scaled continuity of \(\psi\) needs to have \(p'(x)g\) replaced by \(\varepsilon p'(x)g\) for some \(\varepsilon\) scaling, which may change the total mass. Under steady-state conditions it is easy to account for any such spurious changes.

We remark that the standard microscopic model \[2, 3, 6\] replaces (4b) with \(p' = P^s\), omits (5), and to be consistent needs to have \(p'(x)g\) replaced by \(\varepsilon p'(x)g\) in (3b) and (4a). The novel expression (4b)-can be viewed as a scaled continuity of \(\psi\) needs to have \(p'(x)g\) replaced by \(\varepsilon p'(x)g\) for some \(\varepsilon\) scaling, which may change the total mass. Under steady-state conditions it is easy to account for any such spurious changes.

We remark that the standard microscopic model \[2, 3, 6\] replaces (4b) with \(p' = P^s\), omits (5), and to be consistent needs to have \(p'(x)g\) replaced by \(\varepsilon p'(x)g\) in (3b) and (4a). The novel expression (4b)-can be viewed as a scaled continuity of \(\psi\) needs to have \(p'(x)g\) replaced by \(\varepsilon p'(x)g\) for some \(\varepsilon\) scaling, which may change the total mass. Under steady-state conditions it is easy to account for any such spurious changes.

We remark that the standard microscopic model \[2, 3, 6\] replaces (4b) with \(p' = P^s\), omits (5), and to be consistent needs to have \(p'(x)g\) replaced by \(\varepsilon p'(x)g\) in (3b) and (4a). The novel expression (4b)-can be viewed as a scaled continuity of \(\psi\) needs to have \(p'(x)g\) replaced by \(\varepsilon p'(x)g\) for some \(\varepsilon\) scaling, which may change the total mass. Under steady-state conditions it is easy to account for any such spurious changes.

We remark that the standard microscopic model \[2, 3, 6\] replaces (4b) with \(p' = P^s\), omits (5), and to be consistent needs to have \(p'(x)g\) replaced by \(\varepsilon p'(x)g\) in (3b) and (4a). The novel expression (4b)-can be viewed as a scaled continuity of \(\psi\) needs to have \(p'(x)g\) replaced by \(\varepsilon p'(x)g\) for some \(\varepsilon\) scaling, which may change the total mass. Under steady-state conditions it is easy to account for any such spurious changes.

We remark that the standard microscopic model \[2, 3, 6\] replaces (4b) with \(p' = P^s\), omits (5), and to be consistent needs to have \(p'(x)g\) replaced by \(\varepsilon p'(x)g\) in (3b) and (4a). The novel expression (4b)-can be viewed as a scaled continuity of \(\psi\) needs to have \(p'(x)g\) replaced by \(\varepsilon p'(x)g\) for some \(\varepsilon\) scaling, which may change the total mass. Under steady-state conditions it is easy to account for any such spurious changes.
MODELING OF THREE DIMENSIONAL UNSTABLE MISCIBLE DISPLACEMENT IN POROUS MEDIA

Mary Fanett Wheeler  
Department of Mathematical Sciences  
Rice University  
Houston, TX 77251-1892  U.S.A.

and

Ashok Chilakapati and Clarence Miller  
Department of Chemical Engineering  
Rice University  
Houston, TX 77251-1892  U.S.A.

Abstract Simulation of three dimensional unstable miscible displacement utilizing a numerical method which combines the modified method of characteristics is described. Numerical experiments are presented.

I. INTRODUCTION

In flow through a porous medium small flow disturbances are continuously generated due to heterogeneities of the medium, both physical and chemical. The processes of finger growth and interaction are linked to the heterogeneity of the medium and in particular to the spatial variation of porosity and permeability. For example viscous fingering arises in the displacement of a fluid in a porous medium by a less viscous fluid. That is small perturbations tend to grow with time producing large protrusions or fingers. A large number of miscible enhanced oil recovery processes are dominated by viscous fingering and cause a severe reduction in displacement efficiency. In groundwater fingering can cause uneven spread of a contaminant. Chemical heterogeneities arising from adsorption and ion exchange, further accentuate the instabilities.

Numerical simulation is a major tool in understanding the effects of various parameters on miscible viscous fingering. Investigation of the effect of the structure of the porous medium has been an active research area for many years. A major difficulty is in developing realistic descriptions of the medium for numerical simulation. Conditional simulation [4, 5, 7, 8, 1] involves the generation of synthetic porous media that are compatible with the available statistical information. To obtain a meaningful statistical result, many realizations must be carried out and then averaged. A number of statistical techniques that have been employed for generating realizations of a porous medium are described in [6].

In [7, 8] Moisiss, Miller, and Wheeler studied by numerical simulation the effects of spatial variation in permeability and of the viscosity ratio on horizontal unstable miscible displacement. Their numerical experiments were limited to a two-dimensional porous media and to a linear (rectangular) geometry. In this paper we briefly discuss extensions of these results to three spatial dimensions. For a linear flood problem we formulate a numerical method combining a Galerkin characteristics method with continuous trilinear elements for the concentration of the invading fluid and a mixed finite element (cell-centered finite differences) for the pressure equation. Some numerical results are presented which involve a porous medium previously studied [3, 7].

II. METHODOLOGY

2.1 The Problem.

Consider incompressible miscible displacement in a porous medium having the shape of a rectangular parallelepiped. The pore space of the medium is initially filled with the resident fluid. The medium is flooded at the side \( x = 0 \) with pure invading (or displacing) fluid. Thus \( x \) is the principal flow direction and \( y \) and \( z \) are the directions transverse to the flow. Let \( L_x, L_y, \) and \( L_z \) be the \( x, y, \) and \( z \) dimensions of the parallelepiped and let \( \Omega = (0, L_x) \times (0, L_y) \times (0, L_z) \). Under the above assumptions, the displacement can be modeled by the following set of equations:

\[
\begin{align*}
\mathbf{u} &= -\frac{k}{\mu} \nabla p \\
\nabla \cdot \mathbf{u} &= 0 \\
\frac{\partial}{\partial t} (\phi c) + \mathbf{u} \cdot \nabla c &= \nabla \cdot \mathbf{D} \nabla c
\end{align*}
\]

Equations (1) (2) can be combined to yield the pressure equation and (3) is the concentration equation. Here \( \mathbf{D} \) is the diffusion tensor. The eigenvectors of this tensor are orthogonal with one of the eigenvectors being \( \mathbf{u} \) and the eigenvalues \( \alpha_1, \alpha_2, \) and \( \alpha_3 \).

We impose the Dirichlet boundary conditions \( p = \rho_0(t) \) at the inflow boundary \( x = 0 \). The outflow boundary conditions at \( x = L_x \) are

\[
\begin{align*}
p &= 0 \\
\frac{\partial c}{\partial t} + \mathbf{u} \cdot \nabla c &= 0
\end{align*}
\]

The remaining faces have no-flow boundary conditions. The initial condition is \( c = 0 \). The viscosity \( \mu \) in (1) is a function of the concentration \( c \) of the invading fluid. The equation of state used in this work is

\[
\frac{\mu}{\mu_r} = [1 + (M^{1/4} - 1)c]^{-4}
\]

where \( \mu_r \) and \( \mu_t \) are the viscosities of the resident fluid and the invading fluid respectively and \( M = \mu_t/\mu_r \).

2.2 Numerical Method

The system of coupled equations is numerically approximated by a predictor corrector time stepping method which combines the modified method of characteristics for the concentration equation and a mixed finite element method (cell centered finite difference method) for the pressure equation (MMOC-MFE).
Let $M_h(Z_h)$ denote the finite dimensional space spanned by the continuous (discontinuous) trilinear (constants) defined on $\Omega$. Let $V_h = V_h(x) \times V_h(y) \times V_h(z)$ where $V_h(x)$ denotes the tensor product space of continuous piecewise linears on $[0, L_x]$ and discontinuous piecewise constants on $[0, L_y] \times [0, L_z]$. $V_h(y)$ and $V_h(z)$ are defined similarly.

Let

$$V_h(z) = V_h(z) \cap \{v|v(x, y, 0) = v(x, y, L_z) = 0\} \quad (7)$$

Similarly we define $V_h(y)$.

We define the inner product $\langle v, u \rangle = \int_0^{L_x} \int_0^{L_y} p(y, z, t^n) v(0, y, z) dy dz$, $v \in V_h \quad (8)$

$$\nabla \cdot U^n, w = 0, \quad w \in Z_h \quad (9)$$

where $C^{0,0} = C^0$.

For $n \geq 0$, the MFE approximation $(P^n, U^n)$ is defined by:

$$<\frac{\mu(C^{n,0})}{\kappa} U^n, \nabla v > - < P^n, \nabla \cdot v > = \int_0^{L_x} \int_0^{L_y} p(y, z, t^n) v(0, y, z) dy dz, \quad v \in V_h \quad (8)$$

$$\nabla \cdot U^n, w = 0, \quad w \in Z_h \quad (9)$$

The combination of the MFEM and the MOC has been previously applied to the solution of miscible displacement problems in two spatial variables [9, 2, 7, 8].

III. NUMERICAL RESULTS

The following data has been used in the simulation. $a_t = 2.76 \times 10^{-3}$, $a_r = 7.73 \times 10^{-5}$ and zero molecular diffusion. The inflow boundary condition for $c$ was unity. We set

$$L_x = L_y = L_z = 1.$$  

The isotropic permeability used in the model is described graphically in Fig.1 for a high and a low permeability planes. Corresponding concentration profiles are shown in Fig.2.

ORDER OF CONVERGENCE ESTIMATES FOR FINITE ELEMENT APPROXIMATIONS OF DEGENERATE PARABOLIC SYSTEMS MODELLING REACTIVE SOLUTE TRANSPORT IN POROUS MEDIA

PETER KNABNER

Universität Augsburg
Institut für Mathematik
Universitätsstr. 8
D-86133 Augsburg, Germany

Abstract - The semidiscrete finite element approximation is studied for a semilinear reaction-diffusion system consisting of a parabolic and an ordinary differential equation, where the nonlinearity is non-Lipschitz, and a related pde of 'porous medium equation' type. Based on stability estimates for the continuous problem, order of convergence estimates are proved for linear elements in energy- and $L^2$-norms, which partially are optimal.

We consider the finite element approximation of equations, which are conceivable as a macroscopic model for transport and adsorption in porous media (cf. [6] for details). A water flow regime, characterized by the water content $\Theta$ and the flux vector field $q$ and assumed to be known, causes the transport of a solute with concentration $u$ by convection and diffusion/dispersion. The substance undergoes a surface reaction with the porous skeleton like adsorption, i.e. there is an adsorbed concentration $v$. The adsorption reaction may be either in non equilibrium, leading to

$$\partial_t (\Theta u) + \rho \Theta u - \nabla \cdot (D \nabla u - q u) = 0,$$

$$\partial_t u = k (\Theta (u) - v) \quad \text{in } Q_T := \Omega \times (0,T), \quad \tag{KA}$$

or in equilibrium, which gives rise to $v = \varphi(u)$, i.e.

$$\partial_t (\Theta u) + \rho \partial_t (\varphi(u)) - \nabla \cdot (D \nabla u - q u) = 0 \quad \text{in } Q_T. \quad \tag{EA}$$

(KA) or (EA) are supplemented by boundary and initial conditions

$$D \nabla u - q u \cdot n = F (\geq 0) \quad \text{on } S_T := S_1 \times (0,T],$$

$$D \nabla u - n \cdot u \cdot n = 0 \quad \text{on } S_T := S_2 \times (0,T], \quad \tag{1}$$

$$u(\cdot,0) = u_0 (\geq 0) \quad \text{on } \Omega.$$

Hereby $\Omega \subset \mathbb{R}^N$ is a bounded domain, $\partial \Omega = S_1 \cup S_2$, such that $q \cdot n \leq 0$ on $S_1$ and $q \cdot n \geq 0$ on $S_2$, $n$ being the outward normal. The nonlinearity $\varphi$, the adsorption isotherm, only fulfills

$$\varphi \in C^0([0,\infty)) \cap C^1_{\text{loc}}(0,\infty) \quad \text{for some } p \in (0,1),$$

$$\varphi(0) = 0, \quad \varphi(s) > 0 \quad \text{for } s > 0, \quad \varphi \text{ is non-decreasing,} \quad \tag{2}$$

such that degeneration may occur at $u = 0$ leading to finite speed of propagation of supp $u$ (and supp $v$) and limited smoothness of the solution. For the following results certain conditions are necessary with respect to the coefficients. These are fulfilled if the rate parameter $k > 0$ in (KA) is a constant and the other coefficients depend only on space satisfying

$$\Theta(x) \geq \Theta_0 > 0, \quad \rho(x) \geq \rho_0 > 0, \quad \nabla \cdot q(x) = 0, \quad \tag{3}$$

$D(x)$ is symmetric and positive definite uniformly in $x$.

One may think of other rate functions, i.e. descriptions of $\Theta$, than the explicit one in (KA), but this specific structure is important for the following considerations. For the semilinear case (KA) we have despite of the possible degeneration the optimal Lipschitz stability of $u$ with respect to the energy norm

$$\|u\|_{Q_T} := \max_t \|u(t,\cdot)\|_{L^2(\Omega)} + \|\nabla u\|_{L^2(Q_T)}.$$

Theorem 1. Let $(u_1, v_1)$ be weak solutions for the data $u_0$, $v_0$ and $F$, then

$$|u_1 - u_2|_{Q_T} \leq C (\|u_0 - u_{01}\|_{L^2(\Omega)} + \|v_0\|_{L^2(\Omega)} + \|F_1 - F_2\|_{L^2(S_{1T})}).$$

Here and in the following $C > 0$ is a constant independent of the quantities, with which it is multiplied. The proof (cf. [2], [4]) consists of three basic steps:

(I) Test the pde with the primitive of $u := u_1 - u_2$,

(II) Test the ode with $u$,

(III) Test the pde with $u$.

These steps will reappear in the proofs of all the following assertions.

For $k \rightarrow \infty$ we expect convergence of (KA) to (EA). Under certain conditions on the data the speed can be estimated (cf. [6] for details of the convergence proof):

Theorem 2. Let $(u^{(k)}, v^{(k)})$ be weak solutions of (KA) for the rate parameter $k$ and let $u$ be the weak solution of (EA) for the same data, then:

$$\|u^{(k)} - u\|_{L^2(Q_T)} \leq C_k^{1/2}.$$

This justifies the kinetic approximation for (EA), i.e. to approximate the solution of (EA) by the solution of (KA) for large $k$.

We now turn to the finite element approximation of (KA), where we only consider the discretization in space. Let $\Omega \subset \mathbb{R}^2$ be a bounded domain with smooth $\partial \Omega$ and $T_0$ a regular triangulation of $\Omega$. The discretization parameter $h$ is given by the maximal diameter of the triangles $T \subset T_0$. To simplify the notation, we ignore $\Omega \Delta \Omega$, $T \cap T_0$, i.e. we treat $\Omega$ as a polygonal domain. Let $p$ be a constant and define

611
\( S_h := \{ \chi \in C(\overline{\Omega}) \mid \chi|_T \text{ is linear for each } T \in T_h \}, \)

\[ (f,g) := \int_{\Omega} fg \, dx. \] (4)

Then the consistent semidiscrete Galerkin approximation is given by \( u_h, v_h : [0,T] \rightarrow S_h \) satisfying

\[ (\partial_t (\Omega u_h), \eta) + \rho(\partial_t v_h, \eta) - (L_h(u_h), \eta) = 0, \] (5)

\[ (\partial_t v_h, \eta) = (k(\varphi(u_h) - v_h), \eta) \text{ for } \eta \in S_h, \ t \in (0,T), \] (6)

\[ u_h(0) = u_0, \ v_h(0) = v_0, \] (7)

where \( L_h = L_h(t) \) is defined as follows: For \( \eta \in S_h: \)

\[ (L_h(u), \eta) := -(D \nabla u - q u, \nabla \eta) - \int_{S_2} q \cdot n \eta \, d\sigma + \int_{S_1} F \eta \, d\sigma. \]

The data are assumed to be sufficiently smooth and \( u_0, v_0 \) are taken as the \( L^\infty \)-projections onto \( S_h \).

Despite of the missing Lipschitz continuity of \( \varphi \) there is an optimal order of convergence estimate for the energy norm:

**Theorem 3.** Let \( (u, v) \) be the weak solution of (KA) and \( (u_h, v_h) \) the consistent semidiscrete Galerkin approximation, then:

1. \( \| u - u_h \|_{L_2(Q_T)} \leq C h \)
2. \( \| u - u_h \|_{L^2(Q_T)} \leq C h^{2p} \).

Hereby \( p \) is the Hölder-exponent of \( \varphi \). □

The proof uses an auxiliary linear problem by freezing the non-linearity at the solution. Let \( (u_t^h, v_t^h) \) denote the Galerkin approximation for this problem. Then optimal convergence results for \( u - u_t^h \) are well-known. The remainder \( u_t^h - u_h \) is investigated by means of the basic steps in the proof of Theorem 1 and reasonings in [1]. It is open, whether also in \( \| \|_{L^2(Q_T)} \) the optimal estimates \( O(h^p) \) holds. Investigations of uniform convergence are in progress.

The consistent approximation is only asymptotically in accordance with the physical picture insofar (9) is not a local equation in space. This is achieved by considering the *semidiscrete* Galerkin approximation with mass lumping, which is defined by (7) and

\[ \partial_t (\Omega u_h), \eta) + \rho(\partial_t v_h, \eta) - (L_h(u_h), \eta) = 0, \] (8)

\[ (\partial_t v_h, \eta) = (k(\varphi(u_h) - v_h), \eta) \text{ for } \eta \in S_h, \ t \in (0,T), \] (9)

where

\[ (f,g)_h := \int_{\Omega} I(f,g) \, dz \text{ and for } u \in C(\overline{\Omega}) \]

\[ I(u) \in S_h \text{ is defined by } I(u)(P) = u(P) \text{ for all nodes } P \text{ of } T. \]

In fact (9) is equivalent with the collocation approach requiring the ode to be fulfilled at the nodes \( P. \) Proceeding as in the proof of Theorem 3 we are lead to an additional term which can be interpreted as the quadrature error. If the solution \( u \) of (KA) would have the nondegeneracy property with \( \alpha = \frac{1}{2-p} \), which roughly states that \( u^{1/\alpha} \) grows linearly away from \( \partial \text{ supp } u \) into

\[ \| u - u_h^{(k)} \|_{L^2(Q_T)} \leq C h^{2p} \]

for \( k = O(h^{-\frac{2}{2p}}). \)

References


Abstract Simulation of advection-diffusion equations modeling solute transport in groundwater by higher order Godunov-mixed element technique is described. In this approach, a higher order method models advection, while the mixed method models diffusion. These methods are especially useful for problems with nonlinearities, as nonlinear adsorption terms. Numerical results for solute transport with instantaneous, etF-advection, while the mixed method models transport in groundwater by higher order Godunov-mixed element technique. Where-R

I. INTRODUCTION

As noted in two recent reports by the United States Environmental Protection Agency [1] and Department of Energy [2], the potential of groundwater at many locations in the United States is being adversely affected by the introduction of hazardous chemicals into the subsurface. Such chemicals include BTX (benzene, toluene, xylene), gasoline, herbicides, and pesticides. Modeling the flow of these types of chemicals through the subsurface has seen increasing interest in recent years.

The flow of contaminants in groundwater is influenced by many factors, including the aquifer characteristics (hydraulic conductivity, porosity, etc.), the presence of microorganisms capable of biodegrading certain compounds, and the chemical process of adsorption. Biodegradation is an important aspect of groundwater flow, as many compounds may be eliminated by natural biodegradation. Moreover, natural biodegradation processes may be enhanced to effectively remove contaminants [1]. Adsorption, which is a retardation/release reaction between the solute and the surface of the porous structure, is also a significant factor in contaminant movement. Adsorption can have the effect of segregating a hazardous compound from the groundwater, and slowing the overall movement of the chemical species.

The author and M. F. Wheeler have developed and tested a numerical algorithm for modeling multidimensional, multicomponent contaminant flow which includes the effects of biodegradation and linear adsorption, see for example [3, 4]. In this paper, we consider one-dimensional flow of a chemical species in groundwater undergoing (possibly) nonlinear adsorption. In an earlier paper [5], the author and M. F. Wheeler described a first-order method for simulating this problem. We will describe here a higher-order extension of this technique and use it to study nonlinear adsorption phenomena in one space dimension.

II. THE MATHEMATICAL MODEL

Let c denote the concentration of a chemical species in solution. We assume a source of solute c = c_0 at x = 0, and assume flow is in the positive x direction. In one space dimension, conservation of mass yields

\[
\dot{c}_x + u \cdot \nabla c + D \nabla^2 c = 0, \quad c(0,0) = c^0(x).
\]

Here, the positive constants \( \phi, u \) (cm/h), and \( D/\phi \) (cm²/h) denote porosity, Darcy velocity, and the sum of the molecular diffusion and mechanical dispersion coefficients, respectively. The term \( \alpha \) represents the amount of solute adsorbed, where \( \rho (g/cm^3) \) is the bulk density. In many cases of physical interest, flow is advection-dominated; that is, \( u \) is much larger than \( D \) multiplied by some appropriate length scale. This causes numerical difficulties, which we will discuss in more detail below.

The term \( \alpha \) is in general heterogeneous, depending on the adsorbent surfaces. Adsorption can be divided into two classes, equilibrium and non-equilibrium. We will only consider the case of equilibrium adsorption. Adsorption is assumed to be in equilibrium when the reaction kinetics occur at a much faster rate than the rate of transport. In this case, \( \alpha \) can be written as (see [6, 7]),

\[
\alpha(c) = K_d c^p, \quad 0 < p \leq 1,
\]

where \( K_d \) (cm³/g) is the distribution coefficient. Note that in this case \( \alpha(c) \) is not Lipschitz continuous at \( c = 0 \) for \( 0 < p < 1 \).

We will assume \( \psi(c) \) is described by the Freundlich isotherm,

\[
\psi(c) = K_d c^p, \quad 0 < p \leq 1,
\]

where \( K_d \) (cm³/g) is the distribution coefficient. Note that in this case \( \psi(c) \) is not Lipschitz continuous at \( c = 0 \) for \( 0 < p < 1 \).

For more details on these models, and the mathematical ramifications, see [7].

III. NUMERICAL APPROACH

Under the assumptions given above, (1) can be written as

\[
\phi c_t + R(c) c_x + \alpha c - D c_{xx} = 0, \quad x > 0, \quad t > 0,
\]

where \( R = \partial R/c \), \( \alpha = \partial \alpha/c \), and \( D = D/\phi \). Let \( \mu_c = c + \alpha c \), and let \( \eta(c) \) be the inverse function, \( \eta(\mu(c)) = c \). Then (4) can be written as a parabolic equation in \( \mu_c \):

\[
\mu_c + \alpha \eta(\mu) c - D \eta(\mu) c_{xx} = 0, \quad x > 0, \quad t > 0.
\]

As mentioned before, this equation poses the difficulties which we will discuss in more detail below. We define \( \alpha(c) \) as nonlinear-Lipschitz at \( c = 0 \), and the flow is generally advection-dominated, thus the solution exhibits sharp fronts. The effects of choosing \( p < 1 \) (as opposed to \( p = 1 \)) are to make the fronts even sharper, and to further retard the flow of the chemical species. In fact, in the limit of zero diffusion with \( p < 1 \), solutions to (5) can exhibit shocks for smooth initial data.

For advection-dominated flow problems such as (5), we have studied the application of higher-order Godunov-mixed methods. This class of methods was formulated and analyzed for nonlinear advection-diffusion equations in [8]. A multidimensional extension of the method is described in [9]. We now describe the application of this algorithm to (5).

We truncate our computational domain to a region \( (0, \bar{x}) \). At the point \( \bar{x} \) we will assume the "outflow" boundary condition

\[
\mu_c + \alpha \eta(\mu) c - D \eta(\mu) c_{xx} = 0, \quad x = \bar{x}.
\]

Let \( u_{x_i} < x_{i+1} < \ldots < x_{i+j} = \bar{x} \) be a partition of \([0, \bar{x}] \) into grid blocks \( B_i = [x_{i+1} - x_{i+1/2}, x_{i+1/2}] \), and let \( x_{i+1/2} \) be the midpoint of \( B_i \). \( x_{i+1/2} = x_{i+1/2} - x_{i+1/2} \) and \( x_{i+1} = (x_{i+1/2} + x_{i+1/2})/2 \). Let \( \Delta t > 0 \) denote a time-stepping parameter, and let \( \mu^* = \mu(x_{i+1/2}) \).

On each grid block \( B_i \), we approximate \( \mu^* \) by a piecewise linear function \( \mu^*_i \), where

\[
\mu^*_i \equiv \mu^*(x) \approx \frac{x - x_{i+1/2}}{h_{i+1/2}} \mu^*_{i+1/2} + \frac{x - x_i}{h_i} \mu^*_{i+1/2}.
\]

\( c^0 \) is approximated by a piecewise constant function \( c^0_i \), where

\[
c^0_i \equiv C_{0i} = \frac{c^0_{i+1/2} + c^0_{i-1/2}}{2}.
\]

In (7),

\[
\mu^*_i \equiv C_{0i} + R(C_{0i})y_i.
\]

Let \( y(c) \) denote the diffusive flux, \( y(c, t) = -\partial \eta(\mu(c)) / \partial \mu \). Applying the mixed finite element method to the diffusion terms in (5) with the lowest order Raviart-Thomas approximating spaces, and using the appropriate quadrature rule, \( y \) is approximated by (see [8]),

\[
\frac{c_{n+1/2}^0 - c_{n-1/2}^0}{h_{n+1/2}} = -\Delta t \sum_{B_i} \int_{x_{i+1/2}}^{x_{i+1/2}} y(c_{n+1/2}^i) \, dx,
\]

for \( j = 1, 2, \ldots, J - 1 \). At the inflow boundary,

\[
\frac{c_{n+1/2}^0 - c_{n+1/2}^0}{h_{n+1/2}} = -\Delta t \sum_{B_i} \int_{x_{i+1/2}}^{x_{i+1/2}} y(c_{n+1/2}^i) \, dx.
\]

We discuss the handling of the outflow boundary condition (6) below. Discretizing (5) we obtain the difference equation

\[
\frac{\mu_c^{n+1,i} - \mu_c^n}{\Delta t} + \frac{\alpha \eta(\mu^{n+1,i}) - \alpha \eta(\mu^{n+1,i+1/2})}{h_{i+1/2}} + \frac{\eta^{n+1,i+1/2} - \eta^{n+1,i-1/2}}{h_{i+1/2}} = 0,
\]

which holds for \( j = 1, \ldots, J - 1 \). In block \( B_j \), we first compute the predictor \( \mu^{p,n+1} \) by

\[
\frac{(P)^{n+1,i} - \mu^{n+1}}{\Delta t} + \frac{\alpha \eta(\mu^{n+1,i+1/2}) - \alpha \eta(\mu^{n+1,i-1/2})}{h_{i+1/2}} = 0.
\]

We define \( C^{n+1,i} = \eta(\mu^{n+1,i}) \), and the diffusive flux at \( \bar{x} \) is approximated by

\[
\frac{C^{n+1,i} \Delta x}{h_{i+1/2}} = -\Delta t \sum_{B_i} \int_{x_{i+1/2}}^{x_{i+1/2}} y(c_{n+1/2}^i) \, dx.
\]

Finally, we update \( \mu^{n+1} \) by an equation of the form (12), with \( j = J \).
The term $\eta(t_{n+1}/2)$ is an approximation to $\eta(t_{n+1}/2)$. We approximate this term by characteristic tracing from the point $(x_{j+1/2}, t_{n+1})$ back to time $t_n$, i.e., we define the characteristic $x(t)$ satisfying

$$x(t) = x_n + \eta'(	ilde{x}_n^2), \quad x(t_{n+1}) = x_{j+1/2}^2$$

(15)

Assuming the CFL constraint

$$\lambda_j^n = \Delta t \frac{\eta'(	ilde{x}_n^2)}{h_j} \leq 1,$$  

(16)

then $x(t)$ crosses the time axis at a point $x = t \in B_j$, where

$$x=t = x_{j+1/2}^2 - \Delta t \frac{\eta'(	ilde{x}_n^2)}{2}.$$  

(17)

The term $\eta(t_{n+1}/2)$ is given by

$$\eta(t_{n+1}/2) \equiv \eta(\tilde{x}_j^2)$$

(18)

Thus, the advective part of (12) is handled fully explicitly. Substituting (9), (10), (11), and (14) into (12), we obtain a nonlinear system of equations in $\tilde{u}_j^{n+1}$, $j = 1, \ldots, J$. Once $\tilde{u}_j^{n+1}$ is determined, $\tilde{p}_j^{n+1}$ is updated by (9).

The last step in the calculation at time $t_{n+1}$ is the computing of the slopes, $\delta p_j^{n+1}, j = 1, \ldots, J$. The slope in the last interval, $\delta p_j^{n+1}$, is set to zero. In the remaining intervals we set

$$\delta p_j^{n+1} = \delta_{\text{lim}} p_j^{n+1} \cdot \text{sign}(\tilde{p}_{j+1}^{n+1} - \tilde{p}_{j-1}^{n+1}),$$  

(19)

where

$$\delta_{\text{lim}} p_j^{n+1} = \min(\{\Delta_+ p_j^{n+1}, 0\}, \frac{\Delta_+ p_j^{n+1} \cdot \Delta_- p_j^{n+1}}{\Delta_+ p_j^{n+1} > 0}, 0),$$  

(20)

Here $\Delta_+ p_j$ is the forward difference $(\tilde{p}_{j+1} - \tilde{p}_j)/h_{j+1/2}$, and $\Delta_- p_j$ is the corresponding backward difference. The point of the procedure (19)-(20) is to compute a piecewise linear approximation which doesn't without introducing new extrema into the approximate solution. Thus, in blocks where the solution already has a local extremum, the slope $\delta p_j$ is set to zero.

IV. NUMERICAL RESULTS

In this section, we study the effect on the solution of varying the exponent $p$. We choose $c_0 \equiv 1$, $\phi = .5$, $R = 1.5$, $u = 2.5 \text{ cm/h}$, and $D = .15 \text{ cm}^2/\text{h}$. The computational domain is $0 < x < 100 \text{ cm}$. The initial condition for all cases is plotted in Figure 1.

We first consider the case $p = .8$. To test the convergence of the scheme, we compare the approximate solutions at time $t = 25$ hours, generated using 50 and 100 grid blocks. As seen in Figure 1, these solutions are very close. In Figure 2, we compare solutions for $p = .5, .8$, and $1$ at $t = 25$ hours. This figure shows that increasing $p$ results in sharper fronts and substantial retardation of the solution, as expected.

In conclusion, the Godunov-mixed method approach described here gives solutions which agree with physical intuition. In future work, we will extend the method to model more physically interesting situations in multiple space dimensions.

REFERENCES


**EFFECTIVE DISPERSION MODELS FOR VISCOUS FINGERING IN HETEROGENEOUS MEDIA**

Richard E. Ewing
Departments of Mathematics, Chemical Engineering, and Petroleum Engineering
University of Wyoming
Laramie, Wyoming 82071

**Abstract**—In order to scale the highly localized behavior of viscous fingering generated by heterogeneous media up to computational and field scales, we must develop techniques to obtain effective parameters for coarse-grid models which match fine-grid simulations. In [1], Russell, Young, and the author presented a coarse-grid dispersion model of heterogeneity and viscous fingering to match fine-grid simulation of miscible displacement processes. They adjusted longitudinal and transverse dispersivities in a dispersion tensor to match recovery curves for simulations of viscous fingering on fine grids. Although they were able to match production from various simulations, they pointed out that permeability averages, variances, and standard deviations alone are not able to determine dispersivities, since the specific permeability distribution in each realization can have significant impact upon the flow and hence the recovery. In [2], Espedal et al. consider similar dispersion models for immiscible, two-phase flow. In this paper we combine these ideas for multiphase and multicomponent flow, using dispersion models coupled with accurate treatment of first-order transport effects for both models. This coupling will be very important for fully compositional models, which possess aspects of each process. The dispersion models are presented for both multiphase and multiphase cases. Then accurate high-resolution numerical simulators are introduced and used as our experimental tool. Numerical results illustrate the success of dispersion models for all these problems.

I. INTRODUCTION

The understanding and prediction of the behavior of the flow of multiphase or multicomponent fluids through porous media are often strongly influenced by heterogeneities or quasi-localized phenomena. Although considerable information can be gained about the physics of multiphase flow of chemically reacting fluids through porous media via laboratory experiments and pore-scale models, the length scale of these data is quite different from that required for field-scale understanding. The coupled fluid/fluid interactions are highly nonlinear and quite complex. The presence of heterogeneities in the medium greatly complicates this flow. We must understand the effects of heterogeneities coupled with nonlinear parameters and functions on different length scales. We use the simulators as "experimental tools" in the laboratory of supercomputer environments to simulate the process on increasingly larger length scales to develop intuition on how to model the effects of heterogeneities and viscous fingering at various levels.

II. DISPERSION MODELS

In order to ensure that the information passed from scale to scale is dependent upon the physical properties of the flow and not upon the numerics of the specific simulator, we have extensively studied the codes used and have shown them to be essentially free of numerical dispersion and grid orientation effects. The codes utilize mixed finite element methods for accurate fluid velocities in the presence of heterogeneities and modified method of characteristics techniques for accurate fluid transport without numerical dispersion.

The miscible displacement of one fluid by another in a porous medium \( \Omega \) is given by

\[
\frac{\partial c}{\partial t} + \nabla \cdot uc - \nabla \cdot D\nabla c = q, \quad x \in \Omega,
\]

where \( c \), a fraction between 0 and 1, is the concentration of the invading fluid, \( \phi \) is the porosity of the medium, \( u \) is the fluid velocity, \( q \) is the flow rate at the wells, \( c \) is the resident concentration at the well and \( D \) is the dispersion tensor given by

\[
D = \phi \left( d_m I + d_t [l(E)] + d_d [d(E)] + d_f [f(E)] \right);
\]

here \( d_m, d_t \), and \( d_d \) are the molecular, longitudinal, and transverse dispersivities, respectively, \( d_f \) is the fluid velocity, \( l(E) \) and \( d(E) \) are the longitudinal and transverse terms.

REFERENCES


Reentry Aerothermodynamic Simulations using the Taylor-Galerkin Finite-Element Method

E. Laurien, M. Böhle, H. Holthoff and J. Wiesbaum

Institute for Fluid Mechanics
Technical University of Braunschweig
3300 Braunschweig, Bienroder Weg 3, Germany

Abstract—The hypersonic flow around space capsule-like bodies under the conditions of reentry into the earth's atmosphere is simulated. As an appropriate numerical algorithm the Taylor-Galerkin finite-element method has been selected. Results are presented for flows in thermodynamical and chemical equilibrium.

I. INTRODUCTION

The engineering need to simulate reentry aerothermodynamics requires the development of new numerical algorithms and their application to realistical configurations. The physical problem involves strong shocks, high temperatures, boundary layers, entropy layers and the interaction of these phenomena. Therefore a numerical method to simulate such flows must posses flexible spatial approximation properties, such as provided by unstructured locally refined computational grids. Furthermore it is desirable to couple the flow computation with computations of the heat flow within the heat-shield consisting of non-ablative ceramic material. We have chosen the Taylor-Galerkin finite-element method [1,2] as the basis for the development of a three-dimensional aerothermodynamical simulation code. In the present paper some computational aspects are outlined and new three-dimensional simulations under reentry conditions are presented.

II. GRID GENERATION

An axisymmetric three dimensional grid around bodies of revolution is generated using the cylindrical coordinate system \( x,r,\varphi \). First a relatively coarse two-dimensional unstructured grid consisting of triangles in the \( x-r \)-plane is computed using transfinite interpolation to generate the nodes and Delaunay triangulation [3] to generate the triangles. The grid is then locally refined in regions of shocks and boundary layers by subdivision of selected triangles with 'hanging nodes' avoided. This grid is then rotated around the axis \( r = 0 \) by small angles \( \varphi_n \) forming an array of \( n \) sectors between successive planes \( 0 \leq \varphi \leq 2\pi \). Corresponding triangles of these planes form skewed prisms, each of which is subdivided into three tetrahedrons in space. By this technique large three-dimensional grids consisting of tetrahedrons can be generated, the tetrahedrons of each plane being geometrically similar. The grid is 'semistructured', i.e. unstructured in \( x,r \) and structured in \( \varphi \).

III. SIMULATION ALGORITHM

As the algorithm to simulate aerothermodynamics of reentry including strong shocks, boundary layers, entropy layers, and thermodynamical and chemical relaxation the explicit two-step Taylor-Galerkin method [1,2] is applied. This algorithm is formulated in a cartesian coordinate system \( x,y,z \) using transformed node-coordinates. Elementwise constant and linear shape functions are used. Beginning with a parallel flow the Navier-Stokes and coupled chemistry equations are integrated in time. Integrals of the shape functions and their derivatives are split into two parts, the first only depending on \( x \) and \( r \) and the second on \( \varphi \). In an efficient implementation of explicit finite-element methods integrals must be precomputed and stored for all permutations of element numbers, local node numbers and coordinate directions. In our method we reduce the computer space greatly by only precomputing the first part for each tetrahedron of one sector and the second for each sector. Using the axisymmetric grid nonaxisymmetric flow can be computed, e.g. with a small angle of attack.

In order to improve shock-capturing properties the algorithm of flux-corrected transport [4,5] has been implemented. A model of chemically reacting air [6] at high temperatures was so far applied to thermal and chemical equilibrium and is currently being implemented.
for nonequilibrium flow. In the equilibrium case the system of chemical reactions as well as the vibrational excitation of the molecules can be decoupled and solved a priory. This precomputed solution is used during the simulation as a Chebychev approximation.

IV. RESULTS

The algorithm is tested using the two-dimensional example of a circular cylinder at an inflow Mach number of 20 for frictionless flow. The computational grid and a perfect gas solution is shown in Fig. 1. A three-dimensional Euler simulation of a sphere (220,000 elements, 40 sectors) is shown in figure 2.

V. CONCLUSIONS

Finite-element simulations of three-dimensional flow around capsule-like configurations can be conducted on axisymmetric semistructured grids. First results for thermochemical nonequilibrium and viscous flow will soon be available. However, physical and numerical models must be validated by comparision with experimental measurements during actual reentries.

VI. REFERENCES


Abstract - We continue our investigation of the three-dimensional evolution of nominally axisymmetric transitional jets subject to axisymmetric, helical or azimuthal perturbations and combinations thereof. Our approach is a computational one, employing inviscid vortex filament techniques to gain insight into the mechanisms leading to the concentration, reorientation, and stretching of vorticity. Our earlier studies had demonstrated the emergence of vortex rings connected by counterrotating pairs of streamwise braid vortices for the case of superimposed axisymmetric and azimuthal perturbations. Furthermore, for the case of a helical perturbation combined with an azimuthal one, we observed the emergence of concentrated streamwise braid vortices all of the same sign. In the present investigation, we study the interaction of two helical perturbations of opposite sign. While they cancel each other at some azimuthal locations, they become amplified at others, thus leading to a complex three-dimensional flow pattern exhibiting regions of strong azimuthal vortices connected by concentrated streamwise vorticity. In addition, the interaction between the opposite-sign helices results in strong azimuthal velocities.

1. Introduction

The present investigation represents a continuation of our earlier studies of the evolution of transitional jets under three-dimensional perturbations (Meiburg, Lasheras and Martin 1989, Meiburg and Martin 1990, Martin and Meiburg 1991). The types of perturbations we have been considering are of wave-like character in the streamwise, helical and azimuthal directions, as past stability analyses had demonstrated their relevance with respect to axisymmetric jets and vortex rings (Batchelor and Gill 1962, Widnall, Bliss and Tsai 1974, and Cohen and Wygnarski 1987, to mention just a few). Recent experimental investigations by Tso and Hussain (1989) as well as Mungal and Hollingsworth (1989) show convincingly that even fully turbulent jets are dominated by ringlike and helical structures whose dynamics become largely independent of the Reynolds number when this dimensionless parameter is large. Furthermore, Corke and Kusek (1990) experimentally demonstrated the possibility of resonance in axisymmetric jets with helical mode pairs. Edwards, Marx and Ashurst (1991) observe large-scale structures of a helical nature in swirling jets as well. Consequently, our series of studies aims at achieving a more complete understanding of the nonlinear growth and dynamics of these structures, with the ultimate goal of successful manipulation and control of jets. Our numerical investigation of axially forced jets emerging from a corrugated nozzle (Martin and Meiburg 1990) showed the formation of vortex rings that set up a strain field with a free stagnation point in the braid region. Small perturbations in the braid vorticity due to the corrugation are subsequently amplified, whereupon pairs of concentrated streamwise counterrotating vortices form in between the vortex rings. This scenario is in accordance with the mechanism suggested by Lin and Curtus (1984) as well as by Neu (1984) for the plane mixing layer. If, on the other hand, the axisymmetric jet is perturbed by a helical wave, a layer of streamwise braid vorticity forms that has the same sign everywhere (Meiburg and Martin 1990). If the helical symmetry is broken by introducing a periodic perturbation in the azimuthal direction, streamwise braid vortices emerge that become amplified in the strain field of the helix. In the present paper, we will study the inviscid evolution and interaction of a helical mode pair of azimuthal wavenumbers +1 and -1. The numerical technique will briefly be described in section 2. In section 3, we will discuss the emerging flow field with particular emphasis on the large-scale vortical structure.

2. Numerical Technique

The non-divergent nature of the velocity field in incompressible flows, along with the definition of vorticity, allows for a complete description of the kinematics of the flow in the form of the Biot-Savart law. Using the theorems of Kelvin and Helmholtz for inviscid dynamics and following the general concepts reviewed by Leonard (1985), vortex filaments are used for the representation of the vorticity field. Each filament is represented by a number of node points along its centerline, through which a cubic spline is fitted to give it a smooth shape. The three-dimensional Biot-Savart integral is reduced to a line integral by assuming an invariant algebraic vorticity distribution around the filament centerline. For the numerical simulation, we limit ourselves to the temporally growing problem, i.e., our flow is periodic in the streamwise direction. We take the velocity difference between the centerline and infinity as our characteristic velocity. The thickness of the axisymmetric shear layer, defined as the velocity jump divided by the maximum slope of the velocity, serves as the characteristic length scale, which results in the filament core radius of 0.5. In these units, the radius of the jet considered here is 5. Hence, the important ratio of jet radius to momentum thickness of the jet shear layer is 22.6. The Biot-Savart integration is carried out with second order accuracy both in space and in time by employing the predictor-corrector time-stepping scheme and the trapezoidal rule for spatial integration, respectively. A more detailed discussion of the numerical method can be found in Ashurst and Meiburg (1988).

3. Results and Discussion

The subject of our investigation is a nominally axisymmetric jet perturbed by two helical waves of azimuthal wavenumbers +1 and -1, respectively. The axisymmetric shear layer is represented by vortex filaments that initially have the form of vortex rings. Consequently, there is no overall swirl in the jet. Numerically, each of the two helical perturbation waves is introduced by slightly displacing the vortex filament centerlines in the streamwise direction. Figure 1 shows the flow field at time t=1.72. The two side views and the streamwise view clearly demonstrate that the two perturbation waves cancel each other near \( z=0 \), whereas they amplify each other around \( z=0 \), thereby forming regions in which the vorticity becomes slightly more concentrated. Thus, a Kelvin-Helmholtz-type instability of the axisymmetric shear layer is triggered near \( z=0 \), which leads to a roll-up of the vorticity layer and to the formation of segments of concentrated vortex rings on opposite sides of the jet. However, these segments are out of phase with each other, so that a contour plot of the azimuthal vorticity component in the plane \( z=0 \) would show a staggered pattern, resembling a Karman vortex street. This phase shift between the emerging vortex ring segments on opposite sides of the jet leads to the interesting situation in which some regions of a vortex filament are convected towards the jet axis, whereas others are displaced away from the axis. In this fashion, the vortex filament, in between the emerging vortex ring sections,
develops a streamwise vorticity component which becomes larger as the amplitude of the Kelvin-Helmholtz instability grows. This tendency has become much more pronounced by time $t=5.70$ (figure 2). We observe the formation of concentrated streamwise vortical structures connecting the out-of-phase vortex ring segments. This situation appears similar to the one in mixing layers with a defect caused by a phase jump at a given spanwise location. By time $t=8.05$ (figure 3), these concentrated streamwise structures have grown in strength, as they encompass an increasing number of vortex filaments. We furthermore see that they spiral around each other where they come together to form a vortex ring segment. This behavior is also clearly visible in the streamwise view, and it becomes considerably more prominent by time $t=11.95$ (figure 4). It creates a strong axial velocity component along the vortex ring segments. This situation locally resembles observations of helical vortex breakdown. Interesting questions concern the dependence of the flow pattern on the ratio of jet radius to jet shear layer thickness, and its stability under additional axisymmetric or azimuthal perturbations. A more detailed investigation along these lines is currently under way.

**Acknowledgements**

This work is supported by the National Science Foundation under grant #CTS-9058065 (to E.M.) and by DARPA under URI contract #N00014-86-K0754 to Brown University. The San Diego Supercomputer Center is providing computer time.
ASPECTS OF THE NUMERICAL COMPUTATION OF HYPERSONIC REACTIVE FLOW

Jean-Antoine Désidéri

INRIA Centre de Sophia Antipolis
2004 Route des Lucioles, 06560 Valbonne - France

General considerations on hypersonic reactive flow and the challenge of computation

The hypersonic flight of a space vehicle when it reenters the atmosphere (high Mach number, up to 30, high angle of attack, up to 30°) corresponds to a critical phase from the viewpoint of aerodynamical control and thermal loads control, and gives rise to the development of advanced numerical specialized simulation tools.

At high altitude, the low-density atmosphere ($p_{\infty}=2.52$ Pa, $\rho_{\infty}=4.28 \times 10^{-5}$ kg/m$^3$ at the altitude of 75 km) undergoes a strong compression across the main detached shock of the external flow. Immediately behind this shock, assuming a freestream Mach number of the order of 30, the temperature reaches several tens of thousand-degrees (K) and air dissociates and is in strong chemical but also vibrational non-equilibrium. Along a particle path, the various non-equilibrium modes relax to equilibrium at different time scales: vibrational equilibrium is first reached, then chemical-equilibrium. The dissociation reactions being endothermic, they absorb an important fraction of the energy; in this process, the temperature decreases rapidly. To realize the quantitative importance of this effect, it is instructive to consider the case of a steady inviscid (external) flow for which, when chemistry is negligible (i.e. at lower Mach numbers, $M_{\infty} < 10$), the temperature at the stagnation point $T_S$ can be related to the freestream temperature $T_{\infty}$ by the relation

$$T_S / T_{\infty} = 1 + \frac{\gamma - 1}{2} M_{\infty}^2$$

which expresses the conservation of total enthalpy per unit mass along a streamline. Trying now this formula with $M_{\infty} = 25$, $\gamma = C_p/C_v = 7/5$ (diatomic gas) and $T_{\infty} = 205$ K (standard atmosphere at an altitude of 75 km) yields $T_S = 25830$ K! In reality, in this inviscid case, air is completely dissociated in the stagnation-point region, and the temperature is (only) near 6000 K, that is, 4 times smaller.

Another important effect of chemistry on a typical blunt-body flow is that it modifies noticeably the shock location; the stand-off distance may be reduced in some cases of 40 % or more; more generally speaking, the entire shock layer is thinner. This effect is also of great concern to the designer, since the intersection of main shock with parts of the structure should be avoided (overheating, destruction).

For all of these reasons, and since laboratory experimentation of this extreme regime is very difficult (and costly) and often impossible today for certain configurations, it is important to develop efficient and validated numerical tools for the prediction of hypersonic reactive flow.

From a numerical point of view, the challenge of this task resides in the necessity of accounting for more complex physical models (more-or-less complete chemistry models, some of which are currently developed on the basis of more recent data; inclusion of wall effects, etc.) and accounting for more complex aerodynamics: e.g. the presence of stronger shocks in particular, which makes the question of robustness more critical and painful for this reason, many authors employ upwind schemes for which the artificial viscosity is inherent to the approximation. Then, the extension of known schemes to the reactive-flow case is usually not immediate, since the diagonalization of the convective terms (Euler terms) in the governing equations depends on the form of the state equation and is different when the fluid is made of several species. Several extensions of the van Leer flux-vector splitting and of the Roe and Osher flux-difference splittings in particular can now be found in the literature.

From the designer point of view, the flows of greatest interest are those in the "near-equilibrium regime". For reasons that will be discussed in the lecture, the numerical discretization of classical type of the equations governing non-equilibrium flow is very stiff in this case, and overcoming this difficulty may reveal the greatest challenge to the computational scientist.

Finally, since shock layers in the hypersonic regime are thinner than in the better-known supersonic regime due to both effects of larger Mach number and chemical dissociation, and since the physical phenomena are more complex (dissociation, vibration, etc.) and more intensive (stronger shocks), it is evident that the control of the quality of the discrete approximation is more critical and is also more strongly dependent on the quality of the employed mesh. Hence, issues such as mesh generation, and more generally, mesh size control become more essential.

Brief description of the lecture

The lecture will emphasize some of the most important aspects of the numerical computation of hypersonic external flow by upwind schemes applicable to arbitrary unstructured meshes.

After the presentation of some general considerations on hypersonic reactive flow, basic equilibrium and non-equilibrium models will be described. The essential effect of the Damköhler number on a typical blunt-body non-equilibrium flow will be briefly discussed.

Hybrid Finite Volumes/Finite-Element upwind schemes will be described for both models. Weakly-coupled (equivalent-)
and strongly-coupled formulations will be compared. Many blunt-body flow computations will be shown to evaluate variously the effect of various parameters, such as the freestream Mach number or the size of the obstacle.

Implicit time-stepping will be presented, and its efficiency demonstrated. The construction of quasi-second-order schemes incorporating monotonicity devices will be introduced with some emphasis on the appropriate choice of the set of physical variables on which limitation should be applied. In particular, the drastic effect on the wall chemical composition of the order of accuracy of the approximation scheme will be shown.

The difficulty to correctly simulate the stagnation-point region in a non-equilibrium flow will be discussed and illustrated.

Many examples of computations will deal with inviscid flow. However, a preliminary assessment of the effect of the transport model on a Navier-Stokes reactive flow computation will be included.

Finally, some considerations on mesh generation will be made. An example will be given in which several meshes have been constructed in the course of the solution convergence. Initially, one constructs a smooth (structured) mesh employing a "hyperbolic grid generator" in which the distribution of cell areas is controlled to target a prescribed external domain boundary. Then, one or more mesh enrichments are made by element division to adapt the (now unstructured) mesh to the solution. The same solver (adapted to unstructured data base) is employed throughout.

References

I. INTRODUCTION

In mathematical physics problems an error of the finite-difference method \( \| \mathbf{u} - \mathbf{y} \| = \| \mathbf{u} - \mathbf{y} \|_1 \) (\( \mathbf{u} \) is the solution of an original differential problem, \( \mathbf{y} \) is the solution of a finite-difference problem) is determined by the number of grid points \( N \) so that as \( N \to \infty \) the equality

\[
\| \mathbf{z} \| = \left[ \frac{1}{N} \right]^m
\]

takes place asymptotically. The value of \( C \) on real "crude" grids may be decreased by redistributing the grid points. Without a priori information on the solution structure the optimal grid can not be constructed. In a general case such information can be obtained when solving the problem numerically, and then the grid can be properly corrected. It is natural to call such a technique the method of grids adaptive to the solution. In this paper this technique is based on minimizing the truncation error \([1]\). In the method the requirement of truncation error minimization leads to the equations for the grid points coordinates. Stationary diffusion-convection problems including those with a small parameter at leading derivative are considered. In such problems the solution has high gradients in the region of boundary layers. Using the difference schemes of the second truncation order in the convective terms is not expedient here due to oscillations of the difference solutions. The first order upwind schemes yield rather low accuracy. In the paper this contradiction is proposed to solve by increasing the upwind scheme accuracy on adaptive to solution grid. One \([2]\) and two \([3]\) dimensional boundary value problems are considered.

II. 1-DIMENSIONAL BOUNDARY VALUE PROBLEMS

Consider the boundary value problem

\[
(L^h\mathbf{y})_i = \frac{1}{h_i} \left( \varepsilon (x_{i+1/2}) \left( \varepsilon (x_{i-1/2}) + \varepsilon (x_{i+1/2}) \right) \right) + \left( \alpha (x_{i+1/2}) \left( \alpha (x_{i-1/2}) - \alpha (x_{i+1/2}) \right) \right) + \left( \alpha (x_{i+1/2}) \left( \alpha (x_{i-1/2}) + \alpha (x_{i+1/2}) \right) \right) - \left( \varepsilon (x_{i+1/2}) \left( \varepsilon (x_{i-1/2}) - \varepsilon (x_{i+1/2}) \right) \right) + \left( \alpha (x_{i+1/2}) \left( \alpha (x_{i-1/2}) + \alpha (x_{i+1/2}) \right) \right)
\]

\[
\left( L^h\mathbf{y} \right) = \left( f + \frac{1}{h_i} \left[ \left( \varepsilon (x_{i+1/2}) \left( \varepsilon (x_{i-1/2}) - \varepsilon (x_{i+1/2}) \right) \right) + \left( \alpha (x_{i+1/2}) \left( \alpha (x_{i-1/2}) + \alpha (x_{i+1/2}) \right) \right) - \left( \varepsilon (x_{i+1/2}) \left( \varepsilon (x_{i-1/2}) - \varepsilon (x_{i+1/2}) \right) \right) + \left( \alpha (x_{i+1/2}) \left( \alpha (x_{i-1/2}) + \alpha (x_{i+1/2}) \right) \right) \right]
\]

Different simplified modifications of grid monitor \((6)\) are possible as well. Going over to the second order simplifies \((6)\) essentially
so that (6) takes the form

$$\lambda_{i+1/2} = \frac{1}{2} \left[ p_{i+1/2} \frac{Y_{i+1} - y_i}{x_{i+1} - x_i} \right] + \alpha \quad (8)$$

III. TWO DIMENSIONAL BOUNDARY VALUE PROBLEMS ON TRIANGULAR GRIDS

Consider the 2D problem

$$\nabla \cdot (c \nabla u) + \nabla \cdot (p \cdot u) = 0, \quad x = (x_1, x_2) \in \Omega, \quad (9)$$

$$u + \mu \frac{\partial u}{\partial n} = \nu, \quad x \in \partial \Omega,$$

where \( p = (p_1, p_2) \) is given vector, which satisfies

$$\nabla \cdot p = 0. \quad (10)$$

The problem (9), (10) may be considered as a model for Navier-Stokes equations. Briefly outline the difference scheme and the derivation of the grid equation for problem (9). They generalize 1D equations (7) and (8).

The standard finite element approximation of (9) on an arbitrary triangular grid may be written for point \( i \) (see fig.1) as

$$\sum_k B_k (y_k - y_i) + \sum_k A_k \frac{y_{k+} - y_i}{2} = 0 \quad (11)$$

In (11) the coefficients \( A_k, B_k \) depend on coefficients \( \varepsilon(x), \mu(x) \) in the original equation and on grid geometry. On the triangular grid with acute angles only coefficients \( B_k \) always are positive. The coefficients \( A_k \) may have any sign, therefore the scheme (11) does not satisfy the maximum principle. Write (11) in the form

$$\sum_k B_k (y_k - y_i) + \sum_k \left( A_k^+ y_{k+} + A_k^- y_{k-} \right) - \frac{1}{2} \sum_k |A_k| (y_k - y_i) = 0, \quad A_k = \frac{A_k^+ + |A_k^-|}{2} \quad (12)$$

Separate equation (12) into two equations

$$(L_h^+ y_i) = \sum_k B_k (y_k - y_i) + \sum_k A_k^+ y_k + A_k^- y_i = 0 \quad (13)$$

$$\frac{1}{2} \sum_k |A_k| (y_k - y_i) = 0. \quad (14)$$

The difference equation (13) is a generalization of 1D upwind difference scheme (2) onto triangular grids. It satisfies the maximum principle. From (14) we may obtain equations for grid points coordinates like (8). In fig.2 the example of adaptive grid for (10) is shown. There are two boundary layers here. The first one is located near the up boundary of width \( \sim \sqrt{\varepsilon} \), the second near the right boundary of width \( \sim \varepsilon \).

IV. GENERALIZATIONS

Note some generalizations of the method. First, we used local optimization of truncation error. Meanwhile it is possible to use optimization of some integral norm of truncation error and to construct the grid equations. Second, in section III we considered the 2D diffusion-convection equation. This approach admits a direct generalization onto such 3D equations. Finally the proposed approach may be extended to time dependent problems. The algorithm and results for 1D evolution problems are given in [4].

References

Invariant Manifold Theorems for the Navier Stokes Equations

S. S. Sritharan
Department of Aerospace Engineering
University of Southern California
Los Angeles, California 90089-1191

Abstract
In this paper we review the results on existence, uniqueness and regularity of invariant manifolds for the Navier-Stokes equations. Some aspects of global attractors are also discussed.

1 Introduction
We will present certain results on the structure of attractors for the Navier Stokes equations. Ladyzhenskaya [11] proved that the time dependent viscous flow in two dimensional bounded domains can be characterized by a compact global attractor defined in the following way. Let \( W(t, 0; \cdot) \) be the solution operator which relates the velocity field at any given time to the data. For the case of fluid flow in two dimensional bounded domains, the operator \( W(t, 0; \cdot) \) is defined in a Hilbert space \( H \) for all positive times as a compact operator and satisfies the semigroup property:
\[
W(t, t_1; W(t_1, 0; \cdot)) = W(t, 0; \cdot), \quad 0 \leq t_1 \leq t.
\]

We then note that the Navier-Stokes equations define a dissipative dynamical system. This means there exists an absorbing set in the solution space inside which all orbits enter after a certain time. Existence of absorbing sets was first noted by E. Hopf in 1941 [9] using energy estimates. Let \( B_R(H) \) be the absorbing ball in the Hilbert space \( H \). Then the global attractor \( \Lambda \) associated with the nonlinear semigroup \( W(\cdot) \) is defined as the \( \Omega \)-limit set of \( B_R(H) \):
\[
\Lambda = \bigcap_{t \geq 0} \bigcup_{\xi \in \Omega} W(t, \xi; B_R(H)).
\]

The set \( \Lambda \) is compact, connected, attracts all bounded sets of \( H \) and is invariant to \( W(t, 0; \cdot) \) for positive as well as negative times. This attractor contains in particular the steady, periodic, quasi-periodic and almost periodic solutions. The Hausdorff dimension \( d_\Lambda \) of \( \Lambda \) has been shown to be finite [1, 12, 14, 21, 22, 3]. Such results for three dimensional flow problems would be of great interest for the understanding of the dynamics of turbulence. Another open problem is the task of proving the existence of global finite dimensional invariant manifolds containing the attractor. The possibility of existence of global invariant manifolds containing \( \Lambda \) is suggested by the theorem of Mane [17] which essentially implies that \( \Lambda \) can be parametrized by \( N_\Lambda \) number of parameters with \( N_\Lambda \geq 2d_\Lambda + 1 \). Coordinates of such manifolds would provide us with an effective means of computing the properties of the attractor (which may be of fractional dimension). In this report we will present certain results in this subject. First result [25] concerns with the hyperbolicity of periodic solutions of the Navier Stokes equations in bounded domains. Analyticity of the invariant manifolds is the central result. For earlier studies on hyperbolicity see [13] [8]. The second result is the upper semicontinuity of the global attractor with respect to regularizations of the Navier Stokes equations by the addition of artificial viscosity terms. We will also present a smoothness result for global invariant (inertial) manifolds for the regularized system. These global (inertial) manifolds can also be regarded as approximate global manifolds for the original system.

In this paper we will only consider viscous flow in bounded domains. For a discussion on various issues of unbounded domains see [25].

2 Governing equations and functional framework
Let \( \Omega \subset \mathbb{R}^n, n = 2 \) or \( 3 \) be a smooth bounded domain. We will consider the problem of finding \( (u, p) : \Omega \times (0, \infty) \rightarrow \mathbb{R}^n \times \mathbb{R} \) such that,
\[
\begin{align*}
u_t + (u \cdot \nabla) u &= -\nabla p + \nu \Delta u & \text{in } \Omega \times (0, \infty), \\
\nabla \cdot u &= 0 & \text{in } \Omega \times (0, \infty),
\end{align*}
\]

\( u(x, t) = u_0(x), t \in \partial \Omega \) \( \times (0, \infty) \) and \( \int_{\partial \Omega} u_0 \cdot ds = 0 \)

and \( u(x, 0) = u_0, \quad \text{for } x \in \Omega. \)

Here if \( \Omega \) is multiply connected then we require that the flux through each component of the boundary be zero.

Let \( (U(x, t), P(x, t)) \) be a basic solution field (which is an orbit on the attractor) that satisfies the boundary conditions. It is known that [11] the orbits on the attractor \( \Lambda \) are regular (that is \( U, P \) are infinitely smooth) and defined for positive as well as negative times. We are interested in studying the solution orbits nearby this given orbit in an appropriate function space. Let us introduce the change of variables \( u = U + v \) and \( p = P + q \) so that \( (u, q) \), satisfy,
\[
\begin{align*}
u_t + (U \cdot \nabla) v + (v \cdot \nabla) U + (u \cdot \nabla) v &= -\nabla q + \nu \Delta v & \text{in } \Omega \times (0, \infty), \\
\nabla \cdot v &= 0 & \text{in } \Omega \times (0, \infty),
\end{align*}
\]

\( v(x, t) = 0 \) \( \text{for } (x, t) \in \partial \Omega \times (0, \infty) \),

\[ \nu(x, 0) = v_0, \quad x \in \Omega. \]

2.1 Functional framework
We will use the vector spaces,
\[
\begin{align*}
\mathcal{J}(\Omega) &= \{ u : \Omega \rightarrow \mathbb{R}^n; u \in C_0^\infty(\Omega), \nabla u = 0 \}, \\
\mathcal{H} &= \{ u : \Omega \rightarrow \mathbb{R}^n; u \in L^2(\Omega), \nabla u = 0, u(x) = 0, x \in \partial \Omega \}, \\
\mathcal{V} &= \{ u : \Omega \rightarrow \mathbb{R}^n; u \in H^1(\Omega), \nabla u = 0; u(x) = 0, x \in \partial \Omega \}.
\end{align*}
\]

624
3 Linear and nonlinear semigroups

3.1 Characterization of the Stokes operator and the semigroup it generates

Let us define the coercive and continuous bilinear form \( a(\cdot, \cdot) : V \times V \to \mathbb{R} \) as

\[ a(u, v) = \int_{\Omega} \nabla u \cdot \nabla v \, dx \]

Let \( g \in H \). Then since \( H \subset V' = \mathcal{L}(V; R) \), by Lax-Milgram lemma there exists \( u \in V \) such that

\[ a(u, v) = (g, v)_{H(0)}, \quad \forall v \in V. \]

This defines the Stokes operator \( A \) as \( Au = g \) with \( u \in D(A) \) and \( D(A) \subset V \) dense. In fact using the Cattabriga regularity theorem \([2, 26, 24]\) we obtain an explicit representation of the domain of \( A \) as \( D(A) = H^2(\Omega) \cap V \) and \( A \) is recognized as

\[ Au = -P_H \Delta u, \quad \forall u \in D(A) \]

where \( P_H : L^2(\Omega) \to H \) is the orthogonal projection. The Stokes operator is self adjoint and positive definite. Moreover \( A \in \mathcal{L}(V; V') \cap \mathcal{L}(D(A); H) \) is an isomorphism on to. We have the continuous, dense and compact embeddings.

\[ D(A) \subset V \subset H \equiv H' \subset V' \subset D(A)' \]

Since \( A^{-1} \) is compact in \( H \) we can define the fractional powers \( A^\alpha, \alpha \in R \subset A \) using its spectral resolution:

\[ A^\alpha u = \int_0^\infty \lambda^\alpha dE(\lambda) u = \sum_{i=1}^\infty \mu_i^\alpha \varphi_i \psi_i, \quad \forall u \in D(A), \]

where \( E(\lambda) \) are the resolution of identity generated by \( A \) and \( \{\mu_i, \varphi_i, \psi_i\} \) are the eigen pair of \( A \). The domain of \( A^\alpha \) is

\[ D(A^\alpha) = \{ v = \sum a_i \varphi_i \lambda^i \lambda_0^2 \lambda_1^2 < \infty \}. \]

It follows from a theorem of Lions \([15]\) that \( D(A^{1/2}) = V \). We will denote by \( A^\alpha \) the dual of \( D(A^\alpha) \) (closure of \( H \) under the norm of \( D(A^{1/2}) \)). The resolvent of \( -A \) satisfies

\[ \| R(\lambda; -A) \|_{L(H,H)} \leq \frac{1}{|\lambda| + \epsilon} \text{ for } \lambda \in \Sigma, \quad 0 < \epsilon < \mu_1. \]

Here \( \Sigma \) is a sector containing the right half plane.

**Lemma 1** \(-A\) generates a compact semigroup that is holomorphic in \( \Sigma \). Moreover for \( 0 \leq \beta \leq \alpha \),

\[ \| S(t) \|_{L(D(A^\beta); D(A^\beta))] \leq \frac{C}{t^{\alpha-\beta}}, \quad t > 0 \]

and \( \exists \epsilon > 0 \) such that,

\[ \| S(t) \|_{L(H,H)} \leq \epsilon \lambda^\alpha, \quad t \geq 0. \]

Here \( \Sigma \) is an acute sector containing the positive real axis.

3.2 Characterization of the inertia terms

Let us now consider the "inertia terms" in equation \((1)\). We will define a linear operator \( L_0 \) and bilinear operator \( B(\cdot, \cdot) \) in the following way.

\[ L_0 u = P_H \left( (U \cdot \nabla) u + (v \cdot \nabla) U \right) \]

and \( B(v, v) = P_H (v \cdot \nabla) v \).

**Lemma 2** If \( U \in A \) then \( L_0 \in \mathcal{L}(D(A^\alpha); D(A^{\alpha - 1/2})) \), \( \alpha \in [0, 1/2] \). The bilinear operator satisfies \( B(\cdot, \cdot) : V \times V \to D(A^{-1/2}) \).

Proof of these results can be found in \([25]\). The second result is actually due to \([6]\).

3.3 The Linearized problem

Let us find \( v \in C(0, \infty; V) \cap C^1(0, \infty; V') \) such that

\[ v_t + L_0 v + Av = 0, \quad t > 0. \]

\[ v(0) = v_0 \in V. \]

This is resolved as

\[ v(t) = S(t)v_0 - \int_0^t S(t-\tau) L_0(v) d\tau. \]

If we define the operator \( K \) as,

\[ [Kv](t) = \int_0^t S(t-\tau) L_0(v) d\tau, \]

then \([I + K]v(t) = S(t)v_0 \). Properties of \( L_0 \) and the estimates on \( S(\cdot) \) allow us to show that for small enough \( T \),

\[ \| K \|_{C(0, \infty; V); C(0, \infty; V')} < 1. \]

Thus we have the convergent series representation.

\[ v(t) = \sum_{n=0}^\infty [K^n S(t)]v_0 = S(t) v_0. \]

Uniqueness of the solution implies that the evolution operator \( Z(t, \tau) \) satisfies \( Z(t, \tau) = Z(t, \eta)Z(\eta, \tau), \quad 0 \leq \eta \leq \eta \leq t \). Using this we extend \( Z(t, \tau) \) to \( 0 \leq t \leq \infty \). Estimates on the Stokes semigroup allow us to prove,

**Lemma 3** For \( \tau < t \) the maps \( t \mapsto Z(t, \tau) \) and \( \tau \mapsto Z(t, \tau) \) are continuous in the uniform operator topology of \( \mathcal{L}(V; V) \cap \mathcal{L}(D(A^{-1/2}); V) \). For \( \tau < t \) the operator \( Z(t, \tau) \in \mathcal{L}(V; V) \cap \mathcal{L}(D(A^{-1/2}); V) \) is compact. We also have

\[ \| Z(t, \tau) \|_{\mathcal{L}(D(A^{-1/2}); V)} \leq \frac{C}{(t - \tau)^{1/2}}, \quad t > \tau. \]

Let us now consider the case where the basic flow is \( T \)-periodic in time. Existence theorem for time periodic solutions for the Navier-Stokes equations can be found in \([25]\). When \( U \) is \( T \)-periodic, \( Z(T, 0) \) is called the monodromy operator and satisfies,

\[ Z(nT, 0) = Z(T, 0)^n, \quad n \geq 1. \]

3.4 Full Nonlinear problem

Let us find \( v \in C([0, T^*(v_0)]; V) \) such that

\[ v(t) = Z(T, 0)v_0 - \int_0^t Z(t, \tau) B(v(\tau), v(\tau)) d\tau \]

\[ v(0) = v_0 \in V. \]

where \( T^*(v_0) \) the maximal time and is determined by the norm of the initial data \( v_0 \). We write \( v(t) = W(t, 0, v_0, t) \). Then we can show that \([25]\).
$v_0 \to W(t, 0; v_0)$ is real analytic in $B_0(0) \subset V$

This means $W(t, 0)$ can be written as a convergent power series

$$W(t, 0; v_0) = \sum_{n=0}^{\infty} \mathcal{H}_n(v_0, \ldots, v_0; t)$$

The n-linear maps $\mathcal{H}_n(\cdot, \ldots, \cdot; \cdot)$ are continuous. We have

$$W(t, 0; v_0) = W(t, \tau; W(\tau, 0; v_0)), 0 \leq \tau \leq t \leq T^n(v_0).$$

For $\Omega \subset R^2$, we have $T^n(v_0) = \infty, \forall v_0 \in V$ [23]. For $\Omega \subset R^2$, $T^n(v_0) = \infty$ if $v_0$ is contained in a sufficiently small ball centered at the origin of $V$ [6]. If $U$ is $T$-periodic in time then $W(nT, 0; \cdot) = W(T, 0; \cdot)^n, \forall n \geq 1$. We note here that the Frechet derivative of this map $W(t, 0; \cdot)$ is, $D W(t, \tau; 0) = Z(t, \tau)$.

4 Invariant cones and manifolds

Let us first state the following fundamental result:

**Theorem 1.** Let the spectrum of the monodromy operator $Z(T, 0) \in L(D(A^0); D(A^0))$ splits into two disjoint sets such that $\sigma(Z(T, 0)) = \sigma_u \cup \sigma_s$ and

$$b = \sup_{\lambda \in \sigma_u} |\lambda| < \inf_{\lambda \in \sigma_s} |\lambda| = a.$$

Let $P_u$ and $P_s$ be the spectral projectors defined by the Dunford's integrals,

$$P_u = \frac{1}{2\pi i} \int_{R_u} R(\lambda; Z(0, T)) d\lambda$$

$$P_s = \frac{1}{2\pi i} \int_{R_s} R(\lambda; Z(0, T)) d\lambda.$$  

(Here $R(\lambda; Z(0, T))$ is the resolvent operator and $\Gamma_u, \Gamma_s$ encircle $\sigma_u, \sigma_s$ respectively). Then $\forall \epsilon > 0$, we can choose a norm in $D(A^0)$, equivalent to the given one such that $\forall v, w \in D(A^0)$

(i) $\|v\|_{D(A^0)} = \|P_u v\|_{D(A^0)} + \|P_s v\|_{D(A^0)}$, $\|P_u\| = \|P_s\| = 1$,

(ii) $\|Z(T, 0) P_u v\|_{D(A^0)} \leq (b + \epsilon) \|P_u v\|_{D(A^0)}$,

(iii) $\|Z(T, 0) P_s v\|_{D(A^0)} \geq (a - \epsilon) \|P_s v\|_{D(A^0)}$.

Hence $P_u + P_s = 1$, $P_u P_s = P_s P_u = 0$ and $P_u, P_s$ commute with $Z(T, 0)$.

This result is in fact valid for any continuous linear operator in a Banach space with the above spectral properties. For the proof of this results see[4]. Let the spectrum of the monodromy operator splits into two sets such that $\sigma(Z(T, 0)) = \sigma_u \cup \sigma_s$, with $\sigma_u \cap \sigma_s = \emptyset$ and $b = \sup_{\lambda \in \sigma_u} |\lambda| < \inf_{\lambda \in \sigma_s} |\lambda| = a$.

**Theorem 2 (Invariant cone theorem).** If $b < 1$, then there exists a double cone that is invariant to $W(T, 0; \cdot)$ locally. Let $K = \{v \in V; \|P_u v\| \leq \frac{1}{2} \|P_s v\|, v \neq 0\}$ and $K = \{0\}$. Then $W(T, 0; \cdot) : B_0(0) \cap K \to K$ and $K$ is injective.

This theorem is proved in [25] using a result on general mappings by Kirchgaessner and Scheurer [10].

**Theorem 3** The invariant manifold theorem Let $b_a, b_u < 1$. Then in a neighborhood $B_r(0) \subset D(A^\alpha), \alpha \in [1/2, 1]$, there exists two unique, analytic manifolds $M_u$ and $M_a$ which are respectively the graphs of the analytic maps $\phi_u : P_u D(A^\alpha) \to P_u D(A^\alpha), \phi_a : P_a D(A^\alpha) \to P_a D(A^\alpha)$ with,

(i) $\phi_u(0) = \phi_a(0) = 0$,

(ii) $D\phi_u(0) = D\phi_a(0) = 0$...tangency condition

(iii) manifolds $M_u$ and $M_a$ are locally invariant under the solution map $W(T, 0; \cdot)$:

$$W(T, 0; M_a \cap B_r(0)) \subset M_a$$

$$W(T, 0; M_u \cap B_r(0)) \subset M_u.$$  

(iv) stable manifold $M_a$ satisfies

$$M_a \cap B_r(0) = \{u \in B_r(0) \text{ such that } W(nT, 0; u) \in B_r(0) \text{ and } n \to \infty\}$$

and $M_u \cap B_r(0) \cap K = \emptyset$.

Unstable manifold $M_u$ satisfies,

$$M_u \cap B_r(0) = \{u \in B_r(0) \text{ such that } W(T, 0; u) \text{ defined } \forall n < 0$$

and tends to zero as $n \to -\infty$ and $M_u \cap B_r(0) \subset K$.

(v) if $u \not\in M_a$ then there exists $\delta > 0$ and $\rho \in N$ such that,

$$\|W(pT, 0; u)\|_{D(A^0)} > \delta.$$

(vi) dist($M_a, W(T, 0; u)$) < dist($M_u, u$) for $u \in B_r(0)$... exponential attraction property of the unstable manifold.

Proof of this result can be found in [25].

5 Regularized system: quest for global unique solvability

In this section we will present the results on a particular regularization of Navier-Stokes equations. For this system, it is possible to prove global unique solvability theorem up to dimension six [16,18,19]. In this regard the regularized system in six dimensions or less, behaves like the two dimensional Navier-Stokes equations.

Let $\Omega \subset \mathbb{R}^n, n \leq 6$ be a smooth bounded domain. Find $(v', p') : \Omega \times [0, \infty) \to \mathbb{R}^n \times \mathbb{R}$ such that,

$$v'(x, t) = 0, (x, t) \in \partial \Omega \times [0, \infty),$$

$$v'(x, t) = 0, (x, t) \in \partial \Omega \times [0, \infty),$$

and $v(x, 0) = v_0, x \in \Omega$.

It is also possible to use periodic boundary condition (periodic in $\mathbb{R}^6$) for this system. Let us define $\hat{V} := \{v \in H^2(\Omega; \nabla u) = 0\}$. Then as in the characterization of the Stokes operator, we can define a positive definite, selfadjoint operator $\hat{A} = P_0 \Delta^2$ with $D(A) = \hat{V} \cap H^4(\Omega)$ then $D(A^{1/4}) = \hat{V}$ as before. We will obtain $B(\cdot, \cdot) : \hat{V} \times \hat{V} \to H$ compared to $B(\cdot, \cdot) : \mathbb{R}^n \times \mathbb{R} \to D(A^{1/4})$. This shows that the nonlinearity in the regularized system has a better behavior. We can prove that,

**Theorem 4** If $\Omega \subset \mathbb{R}^n, 2 \leq n \leq 6$, then for a given $f \in L^2(0, T; \mathbb{R})$ and $\psi \in H, \exists$ a unique solution $v'$ to (9) such that $v' \in L^2(0, T; : \mathbb{R}) \cap C([0, T]; H),$
6 Global invariant manifolds

For the regularized system introduced in section (5) we can prove the existence of a compact global attractor as well as the local invariant manifolds of the type discussed earlier [19],[20]. Moreover, it is also possible to establish the existence of global invariant varieties of the type introduced in (5]. These global invariant varieties are modelled on finite dimensional linear manifolds spanned by the eigenfunctions of \( \hat{A} \). The global invariant varieties defined below have \( C^1 \) smoothness as compared to the Lipschitz manifolds proposed in [5].

Definition 1 \( \mathcal{M} \) is called a \( C^1 \)-inertial manifold if it is finite dimensional Lipschitz manifold whose derivatives are Lipschitz, has compact support, exponentially attractive, contains the global attractor and is invariant to the action of the solution map (the nonlinear semigroup) in the neighborhood of the global attractor.

For the two dimensional regularized system with periodic boundary conditions we have

Theorem 5 Let \( N^* \) be sufficiently large so that the eigenvalues of the operator \( \hat{A} \) satisfy

\[
\mu_{N^* + 1}^{1/2} \geq d_1/\epsilon \quad \text{and} \quad \mu_{N^* + 1}^{1/2} - \mu_{N^* + 2}^{1/2} \geq d_2/\epsilon
\]

(10)

Then, \( \forall N > N^* \), \( \exists C^1 \)-inertial manifolds \( \mathcal{M}_N \) which are graph of maps from \( P_N \hat{V} \) to \( Q_N \hat{V} \), where, \( P_N = (I - Q_N) \) is the projection in to the invariant subspace corresponding to the first \( N \)-eigenvalues of \( \hat{A} \).

Proof of this theorem involves two steps. First we use the general method in [5] to establish the existence of Lipschitz manifolds. The spectral condition 10 is in fact satisfied for this case. We then prove that under the same hypothesis of the theorem, the above manifolds have Lipschitz derivatives [19]. This is proven using methods similar to that in [23] for the analyticity theorem for the stable-unstable manifolds as described in the early part of this paper. These results on the regularized system motivate the Limit case \( \epsilon \to 0 \): Let \( v \) be the Hopf-class weak solution to the Navier-Stokes equations. Then for \( \Omega \subset R^n \), \( n = 2,3 \) we have,

Theorem 6 If \( v' \in L^\infty(0,T;L^1(\Omega)) \) uniformly in \( \epsilon \) then \( \exists C > 0 \) such that for \( \nu > C \) we have as \( \epsilon \to 0 \), \( v' \to v \) strongly in \( L^2(0,T;H) \)

Recall that, If the Hopf-class weak solution to the NS equations \( v \in L^\infty(0,T;H) \cap L^2(0,T;H) \) satisfies \( v \in L^\infty(0,T;L^4(\Omega)) \) then it is unique.

7 Upper semicontinuous global attractors

We will now state the following powerful result [20] regarding such convergence for the two dimensional case. Let us define the global attractor for the regularized system (0) as

\[
\Lambda_\epsilon = \bigcap_{\rho > 0} \bigcup_{\nu > 0} W_\epsilon(t,0:B_\rho(H))
\]

where \( W_\epsilon(t,0:--) \) is the nonlinear semigroup associated with the regularized system and \( B_\rho(H) \) denotes absorbing ball in \( H \) for the semigroup \( W_\epsilon(t,0:) \).

Theorem 7 \( \forall \epsilon \geq 0 \) and \( \forall \nu > 0 \) the global attractor \( \Lambda_\epsilon \) corresponding to the two dimensional regularized system (with periodic boundary conditions) has the following properties.

(i) \( \Lambda_\epsilon \) is compact and attracts bounded sets of \( H \).

(ii) \( W_\epsilon(t,0: \Lambda_\epsilon) = \Lambda_\epsilon, \quad t \in \mathbb{R} \).

(iii) \( \Lambda_\epsilon \) is upper semicontinuous at \( \epsilon = 0 \): \( \delta_H(\Lambda_\epsilon,\Lambda) \to 0 \) as \( \epsilon \to 0 \). Here the semidistance \( \delta_H \) is defined as

\[
\delta_H(\epsilon,\Lambda) = \sup_{u \in \Lambda} \inf_{v \in \Lambda_{\epsilon}'} \| u - v \|_H.
\]

We note that \( \Lambda_\epsilon \) above corresponds to the global attractor for the two dimensional conventional Navier Stokes system.

Proof of this theorem involves certain uniform estimates (independent of \( \epsilon \)) for the solution of the regularized system and an approximation result on the upper semicontinuous global attractors for nonlinear semigroups. The details of the proof can be found in [20].

Remark: In order to show the continuity of \( \Lambda_\epsilon \) at \( \epsilon = 0 \), we need to prove the lower semicontinuity result: \( \delta_H(\Lambda_\epsilon,\Lambda) \to 0 \). This issue is presently open.

Acknowledgement: This research has been supported by the faculty research innovation fund of the University of Southern California and by the Office of Naval Research through the URI Research Contract No: N00014-86-K-0679.

References


PRESENTATION OF A SECOND ORDER TIME SCHEME
USING THE CHARACTERISTICS METHOD AND
APPLIED TO THE NAVIER-STOKES EQUATIONS

K. BOUKIR, B. METIVET.
E. RAZAFINDRAKOTO
Electricité de France
Etudes et Recherches, IMA
1, av. du Général de Gaulle
92141 Clamart-Cedex France

Y. MADAY
Analyse Numérique
Université P. & M. Curie
Tour 55 - 65, 5e étage
4 Place Jussieu
75252 Paris Cedex 05 France

ABSTRACT - We present here an approximation of incompressible 2D or 3D Navier-Stokes equations in velocity-pressure formulation with a scheme which is second order in time and of finite element type in space. The convective part is treated according to a characteristics method, the Stokes part is solved with the help of an Uzawa algorithm.

Theoretical results concerning the precision and the stability are given. Numerical tests show the improvement given by this second order scheme compared to a first order scheme also using div operators.

I. DESCRIPTION OF THE SECOND ORDER TIME SCHEME

This scheme is based on an operator splitting method similar to the one proposed in [1], [2] and [3].

We denote by \( \Omega \subset \mathbb{R}^N (N = 2 \text{ or } 3) \) the computational domain and by \([0, T]\) the time interval. We consider the Navier-Stokes equations: find \( v: \Omega \rightarrow \mathbb{R}^N \) and \( p: \Omega \rightarrow \mathbb{R} \) solutions of

\[
\begin{align*}
\frac{\partial v}{\partial t} + v \cdot \nabla v - \nu \Delta v + \nabla p &= f \quad \text{in } \Omega \times [0, T], \\
\text{div } v &= 0 \quad \text{in } \Omega \times [0, T], \\
v(x, 0) &= v_0(x) \quad \text{for } x \in \Omega, \\
v &= v_d \quad \text{on } \Gamma \times [0, T].
\end{align*}
\]

Let \( \Delta t \) be the time step and \( t^{n+1} = (n+1)\Delta t \). The second order time scheme is defined by two steps.

Concerning the convection step, for any point \( x \in \Omega \), we introduce the characteristic curve \( X^{n+1}_x: [t^n, t^{n+1}] \rightarrow \mathbb{R}^N \), solution of

\[
\begin{align*}
\frac{dX^{n+1}_x}{dt}(t) &= v_x \left( X^{n+1}_x(t) \right), \\
X^{n+1}_x(t^n) &= x,
\end{align*}
\]

where \( v_x = 2 v - v_{n-1} \).

We set \( \tilde{v}_1^{n+1}(x) = v_x \left( X^{n+1}_x(t^n) \right) \), and \( \tilde{v}_2^{n+1}(x) = v_x \left( X^{n+1}_x(t^n - 1) \right) \).

The Stokes step consists in computing the approximations \( v^{n+1}: \Omega \rightarrow \mathbb{R}^N \) and \( p^{n+1}: \Omega \rightarrow \mathbb{R} \) of the velocity and the pressure at time \( t^{n+1} \), solutions of

\[
\begin{align*}
\frac{v^{n+1} - 2 v^n + v^{n-1}}{2\Delta t} - \nu \Delta v^{n+1} + \text{grad } p^{n+1}
\end{align*}
\]

\[
\Delta t
\]

\[
\text{in } \Omega,
\]

\[
\text{div } v^{n+1} = 0 \quad \text{in } \Omega,
\]

\[
v^{n+1} = v_d \left( \cdot , t^{n+1} \right) \quad \text{on } \Gamma.
\]

We denote by \( \| \cdot \|_2 \) the norm of \( H(\Omega) \). Define \( \| u \|_{2} = \| u \|_{2}^{2} + \nu \Delta t\| u \|_{2}^{2} \).

II. THEORETICAL RESULTS

In this section, we only consider problems (1)-(4) the solution \((v, p)\) of which is sufficiently smooth. Moreover, we assume that \( v_d = 0 \).

II.1 Time consistency error

It is easy to see that only the momentum conservation equation produces a non zero consistency error.

For any point \( x \in \Omega \), let \( \chi^{n+1}: [t^n, t^{n+1}] \rightarrow \mathbb{R}^N \) denote the characteristic curve defined by a system analogous to (5), where the discrete velocity is replaced by \( v \). We also set

\[
h_x(t) = v \left( \chi^{n+1}(t), t \right) \quad \text{for } t \in [t^n, t^{n+1}].
\]

The consistency error is of order 2: it is equal to

\[
E(v, p) = \text{Sup} \left\{ \| e(x, n+1) \|, \quad x \in \Omega, \quad n / e^{n+1} \in [0, T] \right\},
\]

where

\[
e(x, n+1) = - (\Delta t)^2 \left[ \frac{1}{3} \frac{d^3 h_x(t^{n+1})}{dt^3} + \frac{\partial^2 v}{\partial t^2} . \nabla v (x, t^{n+1}) \right] + O((\Delta t)^3)
\]

(\( \| \cdot \|_2 \) is the euclidian norm of \( \mathbb{R}^N \)).

II.2 Stability results

The present results are concerned with unsteady flow computations - the steady case is under consideration -.

The case of the linear convection-diffusion problem

\[
\frac{\partial v}{\partial t} + u \cdot \nabla v - \nu \Delta v = f \quad \text{in } \Omega \times [0, T],
\]

\[
v(x, 0) = v_0(x) \quad \text{for } x \in \Omega, \quad v = 0 \quad \text{on } \Gamma \times [0, T],
\]

has been already studied in [1]. We can deduce from Ewing and Russell's convergence results that the scheme is unconditionally stable for the norm \( \| \cdot \|_1 \); moreover we have exhibited an upperbound of the sequence \( \| v^{n} \|_{\nu} \) by using a somewhat different proof.

629
Proposition 1: If \( \text{div } u = 0 \) in \( \Omega \times J_0 \), \( \Gamma \times J_0 \), \( T_l \), there exist constants \( C(u), C(f), C(v^0, v^1) \), depending only on \( \Omega \) and respectively \( u, f, (v^0, v^1) \), such that for \( \Delta t \leq C(u) \), we have
\[
\| v^n \|_{L^2} \leq C(f) \| T + C(v^0, v^1) \|.
\] (8)

In the nonlinear case of the Navier-Stokes equations, we can presently get stability results only when introducing the space discretization. Let \( h \) be the space step and \( k \) the degree of the velocity finite elements. By using techniques similar to those of [1], we can prove the

Proposition 2: There exist constants \( C_1 \) and \( C_2 \) depending only on \( \Omega \) and respectively \( (v, \nu_h, \nu_k) \) and \( (v, \nu_h, \nu_k), e^{T \xi_0 / 2} \) such that, for \( h \) small enough and \( k > N/2 \), the condition \( \Delta t \leq \inf (1/C_1, C_2 h^{N+1}) \) yields to
\[
\| v^h \|_{L^1, \infty} \leq C_1.
\] (9)

The scheme appears slightly less stable than the first order scheme, which is unconditionally stable ([6]), but this instability may be only due to the techniques of proof.

III. NUMERICAL RESULTS

The second order time scheme has been implemented in the thermal-hydraulic finite element code N3S developed at EDF ([4]). Comparisons have been done with the original first order scheme ([4], [5], [6]). In all the cases, the second order scheme improves the results ([7]).

III.1 Results on analytical steady cases

We have observed on analytical steady tests that the consistency error affects more the pressure results than the velocity ones (cf. Fig. 1 to 3 where results are shown for the same mesh and \( \Delta t = 10^{-3} \)). However, for the tested cases, with the first (resp. second) order scheme, the computed time precision is of order 1 (resp. 2) for both velocity and pressure. We get thus a numerical information concerning the pressure rate of convergence since presently there is no theoretical results for any of the two schemes. Moreover we have observed that a too strong refinement of the time step leads to worth results. In particular, when in the momentum equation the convective part is very important compared with the diffusion-pressure part, we cannot get a good precision without refining the mesh together with the time step. The theoretical space-time precision of the second order scheme applied to the Navier-Stokes equations is under consideration (see [6] for the first order one).

III.2 Results on an unsteady case

The computations of the Gamm workshop ([8]) have been performed with the two schemes. They are concerned with a flow in a heated cavity at a zero Prandtl number and with homogeneous Dirichlet boundary conditions. The following result is given as an example of the improvements given by the second order scheme. For a Grashof number equal to 30000, the reference computations predict an unsteady periodic state which is well computed with the second order scheme, although the first order one converges to a steady state.

IV. GENERALIZATION

This scheme can be generalized to a k-order time scheme ([3]). The consistency error has been studied in [7]. Concerning the order 3, the scheme has been succesfully used together with the spectral methods ([2]). The stability results are under consideration.

References


Abstract - For the 2D Navier-Stokes equations as a specific model for dynamical systems, we have found that the chaotic transition is couched on a 2-torus-like manifold which is the product space of a circle and multiply-periodic orbit. This is conceptually similar to the unstable 2-torus of Ruelle-Takens-Newhouse.

I. INTRODUCTION

Although it is agreed that the laminar-to-turbulence transition begins with emergence of a periodic motion from the fixed point (laminar flow), there is no general agreement as to what happens next to the periodic motion as the forcing (Reynolds number) is further increased. Several scenarios leading to chaos have been proposed; the unstable quasiperiodic 2-torus of Ruelle-Takens-Newhouse[1,2], period-doubling of Feigenbaum[3], and intermittency of Pomeau-Manneville[4] (see, for instance, the review article of Eckmann[5]). We shall show that the 2D Navier-Stokes equations in a cyclic domain follow, at least conceptually, the scenario of Ruelle-Takens-Newhouse closer than any others, although the actual state of affairs is somewhat more complicated. Prior to chaos, there appears a multiply-periodic orbit and the chaotic transition takes place on a 2-torus-like manifold which is the product space of a circle along the longitude angle and the multiply-periodic orbit in the plane of latitude angle. Instead of the perturbation of a circle to the equilibrium states, we have found experimentally that (3) admits the following equilibrium:

\[ V_{n}^{c} = \text{coast}, \quad B_{n}^{p} = \text{periodic}, \quad B_{n}^{c} = \text{chaotic}. \quad (a,b,c) \]

And of (a) and (b), (c) must be rationally related, otherwise it becomes the 2-torus. This is shown in Fig 1(b).

II. A SET OF 860 EVOLUTION EQUATIONS

For a 2D periodic flow with no mean flow, it is most expedient to Fourier analyze the velocity field \( \mathbf{U}(x) \) and body forces \( \mathbf{D}(x) \) in a square region of side \( L \) by \( \mathbf{u}(k) = \sum_{q} \mathbf{U}(k) \exp(ik \cdot q) \), where \( k = (2\pi/L) n \), \( n \) is the wavevector. By spanning incompressible \( \mathbf{U}(x) \) and \( \mathbf{D}(x) \) by the unit polarization vector \( \hat{e}_{k} \) normal to \( k \), the incompressible Navier-Stokes equations give rise to the so-called triad-interaction representation in spectral form

\[
\frac{d u_{k}}{dt} = \nu \sum_{q} \mathbf{U}(\hat{e}_{k} \cdot \hat{e}_{q}) u_{k} u_{q} + \mathbf{D}(\hat{e}_{k} \cdot \hat{e}_{q}) \tilde{u}_{k} \tilde{u}_{q}.
\]

where \( \nu \) is the kinematic viscosity, \( k = |k| \) and \( \hat{e}_{k} \) is the symmetrized coupling coefficient. Let us label the wavevectors in the successive k-rings. There are 430 wavevectors for isotropic truncation of the upper wave number \( k \sim 16 \). By abbreviating \( U_{n}^{c} = u_{n}^{c} \) and \( G_{n}^{c} = g_{n}^{c} \), enumeration of (1) yields a set of 430 equations for \( u_{n}^{c} \) with 106,244 triad-interaction terms in the right hand side.

For computation we split \( u_{n}^{c} \) into the real and imaginary parts by \( u_{n}^{c} = u_{n}^{r} + i u_{n}^{i} \). Denote the vector of \( u_{n}^{r} \) and \( u_{n}^{i} \) by \( \mathbf{u}_{n}^{r} \) and \( \mathbf{u}_{n}^{i} \), by \( \mathbf{D}_{n}^{r} \) and \( \mathbf{D}_{n}^{i} \), by \( \mathbf{F}_{n}^{r} \) and \( \mathbf{G}_{n}^{r} \). The set of 860 equations has the vector form

\[
\dot{\mathbf{u}} = -\nu \mathbf{F}(\mathbf{u}) + \mathbf{G}(\mathbf{u}),
\]

similar to the evolution equation of Constantin et al.[9]. The energy and enstrophy conservations follow from \( \langle \mathbf{u} \mathbf{u} \rangle \geq \langle \mathbf{u} \mathbf{u} \rangle \geq 0 \), where \( \langle \cdot \rangle \) is the scalar product.

III. FIVE EQUILIBRIUM STATES

For the present study, we shall restrict ourselves to the single-mode and three-mode forcing

\[
\begin{align*}
V_{1} &= \begin{pmatrix} 0 & \ldots & \varepsilon_{1}^{2} \end{pmatrix}, \\
V_{3} &= \begin{pmatrix} 0 & \ldots & \varepsilon_{1}^{2} \end{pmatrix},
\end{align*}
\]

where \( \varepsilon \)'s are assumed constant. For the independent forcing, trajectory behavior can best be categorized by the amplitude-angle form \( \mathbf{u}_{n} = \mathbf{F}_{n} \exp(i\phi_{n}) \).

Hence, the equations of motion in \( R_{n}^{c} \) and \( \omega_{n}^{c} \)

\[ \mathbf{U}_{n} = \mathbf{U}_{n}(V_{n}^{c}, V_{n}^{p}, V_{n}^{c}), \quad \mathbf{U}_{n} = \mathbf{U}_{n}(V_{n}^{c}, V_{n}^{p}, V_{n}^{c}). \]

represent an assembly of \( \mathbf{u}_{n} \)-oscillators. Here, \( V_{n}^{c} \) and \( V_{n}^{p} \) are the vectors of \( R_{n}^{c} \), \( \omega_{n}^{p} \), \( \omega_{n}^{c} \) are the right-hand sides involving the forcing \( \mathbf{F}_{n} \exp(i\phi_{n}) \) for \( j = 26, 27 \) and 28.

We have found experimentally that (3) admits the following equilibrium:

\[ \mathbf{R}_{n}^{c} = \text{cost}, \quad \mathbf{B}_{n}^{p} = \text{periodic}, \quad \mathbf{B}_{n}^{c} = \text{chaotic}. \quad (a,b,c) \]

Only certain combinations of (4) and (5) give rise to the equilibrium states [6] as follows:

ES I: Fixed Point - eqn (4a) & (5a)
ES II: Circle - eqn (4a) & (5b)
ES III: Torus - eqn (4b) & (5c)
ES IV: Chaos - eqn (4c) & (5e)

For ES III the principal frequencies of \( R_{n}^{c} \) and \( \omega_{n}^{p} \) must be rationally related, otherwise the orbit is unclosed. What is, however, unexpected is ES IV. Let us denote the equilibrium state by \( \mathbf{X} = \mathbf{R}_{n}^{c} \exp(i\theta_{n}) \) without the overhead bar. The combination of (4b) and (5d) is explicitly given by

\[ \mathbf{R}_{n}^{c} = \mathbf{R}^{d} \sin(2\pi f_{n} \exp(i\theta_{n})), \quad \mathbf{R} = \mathbf{R}^{d} \sin(2\pi f_{n} \exp(i\theta_{n})). \]

Here, \( R \) is modulated by a sine with amplitude \( \Delta \), frequency \( f_{n} \), phase \( \varphi \), and \( \omega \). \( f_{n} \) and \( \varphi \) are defined similarly for \( \Omega \). The angular velocity \( \Omega \) is measured positive/negative for counter-clockwise/clockwise rotation. Since \( \mathbf{X} = \mathbf{R}^{d} \sin(2\pi f_{n} \exp(i\theta_{n})) \), the phase plot of \( X_{1} \) vs \( X_{2} \) has rotated counterclockwise by about 50° in Fig 1(b). Note that \( R = R_{0}^{d} \sin(2\pi f_{n} \exp(i\theta_{n})) \) and \( \omega = \omega \sin(2\pi f_{n} \exp(i\theta_{n})) \) form a closed curve C (ES III).

Fig 1. Rotating phase plot. (a) R=0.05+0.15sin2\( \pi \left( 0.0127t - 0.3 \right) \), \( \Omega = 2\pi \left( 0.0001t - 0.04 \right) \) over \( t = 1500 \); (b) Phase plot; (c) Construction of 2-torus.
Hence, (6) represents a 2-torus-like manifold (ES IV) of the product space \( C^2 \times S^1 \). For definiteness, we shall say that the latitude angle \( \Omega_1 = 2\pi \theta_1 \) encloses \( C^2 \) and longitude angle \( \Omega_2 = \phi_2 \) revolves \( S^1 \) around the \( S^1 \) (see, Fig 165 of Arnold[10]). By decoupling the two angles, one can generate a segment of Arnold[10].

\[ H_{\text{ref}} \]

Hence,
\[ 4(a) \] and then each loop overlaps periodic orbit to chaos. To be oscillatory, we evolve continually from a multiply periodic orbit on the plane of latitude angle. However, with the emergence of longitude angle there forms a 2-torus-like manifold, similar to the quasiperiodic 2-torus of Ruelle & Takens[4] (see, Fig 1.14 of Ref [11]). Hence, a generalization of Chirikov's[8] area preserving map may be appropriate for ES IV

\[ R_{\text{ref}} \]

One can schematize the chaotic transition as shown in Fig 6. Prior to chaos, a multiply-periodic orbit lies on the plane of longitude angle. However, with the emergence of longitude angle there forms a 2-torus-like manifold, similar to the quasiperiodic 2-torus of Ruelle & Takens[11] (see, Fig 1.14 of Ref [11]). Hence, a generalization of Chirikov's[8] area preserving map may be appropriate for ES IV

\[ \theta_{1,2}, \phi_{1,2} \]

Here, we indicate only the second harmonic of strength \( \omega_2 \) in the absence of which (7) is the original map of Chirikov[8]. It further reduces to the circle map \( \psi(\theta) = \theta + \beta \phi_2 \sin 2\theta \) that Shenker[6] and Rand et al.[7] have investigated as a model for chaotic transition of quasiperiodic orbit on a 2-torus.

**References**

Abstract—This paper proposes a new method to improve the stability condition of difference schemes of a parabolic equation. Necessary and sufficient conditions of the stability of this new method are given and proved. Some numerical examples show that this method has some calculation advantages.

I. Introduction

For the parabolic equation,
\[
\left\{ \begin{array}{l}
\frac{du}{dt} = f\left( x, y, t \right), \quad (x, y, t) \in \Omega, \quad t < T \leq T \\
u(x, y; t) = 0, \quad (x, y, t) \in \Gamma, \quad 0 < t \leq T \\
u(x, y; 0) = \varphi(x, y), \quad (x, y) \in \Omega
\end{array} \right.
\]
(1)

where
\[
\Omega = \{ (x, y) | 0 < x < 1, 0 < y < 1, 0 < t < 1 \}, \quad \tau = \Delta t,
\]
a is a positive constant, many feasible difference schemes were constructed on difference computation of the initial and boundary problem by many authors. When we use and explicit difference scheme to compute the problem (1), the step is limited greatly in order to satisfy stability conditions.

This paper proposes a new method that a fixed space step and alternative time steps are adopted in an arbitrary choosing explicit difference scheme to improve stability conditions for explicit difference schemes which are used to solve the parabolic equation (1.1). According to general idea, stability conditions of the explicit scheme should be satisfied on each time level. As a matter of fact, it is not necessary. When we adopt suitable stable and unstable schemes alternately, then it finally leads to a stable scheme. In this paper we give and prove the improved stability conditions which are much better than the stability conditions of the classical explicit scheme. Therefore, we can increase the time step in practical calculation, finish calculation work in less time and bring calculation advantages of explicit schemes into full play.

II. Sufficient and Necessary Condition of Stability

We can use a lot of kinds of explicit schemes to calculate the equation (1). Here we use the simplest classical explicit difference scheme to calculate the problem (1), and then improve stability conditions step by step.

The classical explicit difference scheme of the equation (1) is
\[
\begin{align*}
\frac{U_{i,j}^{n+1} - U_{i,j}^n}{\tau} & = \frac{a}{h^2} \left( U_{i+1,j}^n + U_{i-1,j}^n + U_{i,j+1}^n + U_{i,j-1}^n \right) \\
& + \frac{a}{h^2} \left( U_{i+1,j}^{n} + U_{i-1,j}^{n} - 6U_{i,j}^{n} \right) + f_{i,j}^n, \quad i + j = \text{odd number}.
\end{align*}
\]
(2)

Then we consider alternative explicit difference scheme of the problem (1),
\[
\begin{align*}
\frac{U_{i,j}^{n+1} - U_{i,j}^n}{\tau} & = \frac{a}{h^2} \left( U_{i+1,j}^n + U_{i-1,j}^n + U_{i,j+1}^n + U_{i,j-1}^n \right) \\
& + \frac{a}{h^2} \left( U_{i+1,j}^{n} + U_{i-1,j}^{n} - 6U_{i,j}^{n} \right) + f_{i,j}^n, \quad i + j = \text{even number}.
\end{align*}
\]
(3)

The sufficient and necessary condition of scheme (4) is
\[
r < \frac{1}{3}
\]
(5)

Finally, in this paper we adopt a fixed space step h and use time steps \( r_1 \) and \( r_2 \) alternately. If we use \( r_1 \) in odd time levels and \( r_2 \) in even time levels and two levels are considered as a whole, the following conclusion for improving stability conditions can be obtained.

Theorem 1. To the equation (1), when the time steps \( r_1 \) and \( r_2 \) are alternately used to difference scheme (4), the sufficient and necessary conditions of stability are
\[
\begin{align*}
R_1 & \geq 6R_2, \quad \text{and} (6) \quad R_1 \geq 6R_2, \\
R_1 - 16R_2^2 - 576R_2^4 & < 0
\end{align*}
\]
where \( r_1 = \frac{\Delta t}{h} \), \( r_2 = \frac{\Delta t}{h^2} \), \( R_1 = r_1 + r_2 \), \( R_2 = r_1r_2 \).

Proof. First, we prove that the sum of the areas (6) and (7) is the necessary stability conditions.

When the time steps \( r_1 \) and \( r_2 \) are alternately used to the homogeneous scheme of difference scheme (4) and two levels are combined into a whole, a new augmented matrix is
\[
C = \begin{bmatrix}
1 - 6r_1 & 2r_1 C \\
2r_1 & 1 + 4r_2 C & 1 + 4r_2 C & 0 \\
1 + 4r_2 C & 1 + 4r_2 C & 1 + 4r_2 C & 0 \\
1 & 2r_1 C & 1 & 2r_1 C \\
\end{bmatrix}
\]
(7)

where
\[
C = \begin{bmatrix}
d_1 & d_2 \\
0 & d_3 \\
\end{bmatrix}
\]
(8)
\[ C = \cos \theta \mathbf{h} + \cos \theta \mathbf{h} + \cos \theta \mathbf{h} \]

\[ a = \left( 1 - 6 r \right) \left( 1 - 6 r \right) + \frac{4 r C^2}{1 + 6 r} \]

\[ b = \frac{4 r C^2 \left( 1 - 6 r \right)}{1 + 6 r} + \frac{2 r C (1 + 4 r C^2) (1 + 4 r C^2)}{1 + 6 r} \]

\[ C^2 = \frac{2 r C (1 - 6 r) \left( 1 - 6 r \right) + \frac{2 r C (1 - 6 r)}{1 + 6 r}}{1 + 6 r} \]

\[ d = 2 r C (1 - 6 r) \left( 1 - 6 r \right) + \frac{1 + 4 r C^2}{1 + 6 r} \]

The characteristic equation of the matrix (7) is

\[ \lambda^2 - (b^2 + b)^2 + a \lambda + a = 0 \]  

(3)

The sufficient and necessary conditions for the roots of quadratic equation (8) to be \( |\lambda| \leq 1 \) are

\[ a + b < 1 + a b - c d < 2 \]  

(9)

By considering the inequality of (9), the areas which satisfy Von-Neumann conditions is.

In the following we will verify that (9) is the sufficient stability condition of scheme (6).

For eigenvalue \( \lambda \) of the matrix \( G \), there is a corresponding unit eigenvector \( e_i \). Then from \( e_i \), we can get an orthogonal basis \( e_i e_j \) in \( C^3 \) space. Set \( U = (e_i e_j) \), it is easy to know that \( U \) is a unitary.

Since eigenvalues of similar matrices are same, we have

\[ G^* = U \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} U^* \]

\[ \| G \|_2 \text{ is a uniformly bounded number when } R \text{ and } R \text{ satisfy the condition (6). We denote this bounded number by } M. \]

Since \( M = \| G \|_2 \), we have

\[ (\lambda_1, \lambda_2) \leq \left( \frac{1}{6 r}, \frac{1}{1 + 6 r} \right) < 1 \]

And \( \| G \|_2 \leq \frac{M}{1 - M} \)

\[ \| G^* \|_2 \leq 2 + \frac{M}{1 - M} \]

\[ \| G^* \|_1 \text{ is a uniformly bounded number on } a \text{ and } n. \]

The whole theorem is proved.

Since the stability condition of the implicit-explicit scheme with explicit form is \( \tau \leq 1/3 \), we can know that it is stable only it \( \tau_1 + \tau_2 < 2/3 \) when a uniform time step is used. Above discussion shows if \( R \text{ and } R \text{ (correspondent } \tau_1 \text{ and } \tau_2 \text{) are chosen to satisfy condition (6), we will get a stable difference scheme. From (6), we can see } \tau_1 + \tau_2 < \frac{2}{3} \text{ in general.} \]

III. Numerical Example

Example: Consider the homogeneous problem of initial and boundary value problem (1).

\[ \phi(x, y, z) = 64y(x - 1)(y - 1) - z, \]

We use difference scheme (6) to calculate approximate solutions. We use the mesh ratio \( r \) when \( k \) is a odd number and we use the mesh ratio \( r \) when \( k \) is an even number.

<table>
<thead>
<tr>
<th>Table 1: Values of Difference Solution</th>
<th>( u_{1,1,1}^N )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>0.415</td>
</tr>
<tr>
<td>4</td>
<td>0.841</td>
</tr>
<tr>
<td>6</td>
<td>1.167</td>
</tr>
<tr>
<td>8</td>
<td>1.365</td>
</tr>
<tr>
<td>10</td>
<td>1.435</td>
</tr>
</tbody>
</table>

Because of the symmetry of the initial value and calculating scheme in this example, the difference solution is also symmetric in \( x, y \), and \( z \) directions. The table 1 only shows a quarter of the numerical values in \( x = 1, y = 1 \). From the table 1, we can see that the difference solutions are convergent.

In the following we use the alternate explicit scheme (2) to calculate the same example. We still choose the same step \( h = 1/20 \), and a mesh ratio be the uniformly largest stable ratio \( r \). So the time step is \( \tau = \frac{h^2}{3 a} \). The computational results are shown on table 2.

<table>
<thead>
<tr>
<th>Table 2: Values of Difference Solution</th>
<th>( u_{2,1,1}^N )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>0.487</td>
</tr>
<tr>
<td>4</td>
<td>0.922</td>
</tr>
<tr>
<td>6</td>
<td>1.262</td>
</tr>
<tr>
<td>8</td>
<td>1.476</td>
</tr>
<tr>
<td>10</td>
<td>1.549</td>
</tr>
</tbody>
</table>

By the comparison of table 1 and table 2, we know, the difference solutions of the eighth level in time direction which are solved by the method proposed in this paper have faster convergence than the difference solutions of the thirty-second level in time direction which are solved by the alternate explicit scheme (2), i.e., the time step of the new method can be chosen four times more than the largest stability time step of the old method. So we only spent one-fourth the computational amount as before to get the same result in the situation of no raising computational complexity.

Remark. The method proposed in this paper can be used to any explicit scheme to improve stability conditions.
Abstract: To design an hypersonic aircraft by experimental way only, would be unacceptable because of the difficulties and so of the induced cost. We have to rely on a mix of theoretical (CFD) and experimental approaches. The methodology of design with CFD rely on check-points based on levels of software and on levels of validation of the codes used in design. major targets of preliminary CFD effort is to demonstrate to the customer that one has confidence in computer tools and, so, in the results of computation. Actually CFD is the only way to build complete engine integration, to share responsibilities between cooperants and to clarify interface problems. On air intakes the effort has to cover performance evaluation: prediction of flow field quality (steady and unsteady), and effects of start and unstart of supercritical flows.

Introduction

The higher the Mach number is, the higher will be the necessity of integration of the shapes due to the smaller area between the bow shock-wave and the vehicle. The use of CFD may afford a decisive progress if its analysis capabilities are sufficient. It is only with such a CFD integration that hypersonic air intakes can be efficiently designed. On one hand we will present CFD requirements to better understand the air intake/air frame interactions, on the other hand we will propose a methodology of air intake integration by CFD, supporting by an example.

CFD Requirements and checking

A complete capability is required for global evaluation of performance and for analysis. Such an analysis is better done with reduced CFD effort on not too complex configuration and on elementary box devoted to some phenomena that have to be mastered. But for global evaluation, we need the complete, hence highly complex geometry and the ability to cope with the more complex physical modelling (turbulence + real gases). The requirements can be summarized in the following table:

Table 1: Requirements for CFD

<table>
<thead>
<tr>
<th>A</th>
<th>Physical modelling level required</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Basic phenomena to be included</td>
</tr>
<tr>
<td>C</td>
<td>Basic geometry/boundary conditions to be handled</td>
</tr>
</tbody>
</table>

- Physical modelling (turbulence + real gases)
- Laminar-transitional-turbulent flows
- Shock-boundary layer interaction problems
- Shock-shock crossing and shock impingement
- Turbulent viscous interactions problems
- Unsteady complex flow with separation
- Reattaching flows
- Vortical flows
- Basic geometry/boundary conditions to be handled

Table 2: Requirements for code assessment

<table>
<thead>
<tr>
<th>A</th>
<th>Physical properties as state equation, viscosity ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Multidomain approach, including analytical</td>
</tr>
<tr>
<td>C</td>
<td>Fluid problems</td>
</tr>
</tbody>
</table>

Methodology of air intake integration

Taking into account the flight envelope in Mach number, angle of attack and sideslip, it is clear that mass flow adjustment, to avoid any external perturbation becomes an untractable task. The major part of CFD effort will be devoted to out of adaptation computations. However the basic adaptation remains a design way to reduce the bad interactions. So the current methodology will go from fitting the geometrical shapes to some design points to making an evaluation of an operation out of adaptation.

An efficient way of making analysis of the forces building is to deduce from the area distribution along X-axis and from the drag distribution an equivalent Cp ratio (the drag variation to the section variation ratio). The integral of this curve (CpA) gives the evaluation of the thrust or drag of a given
configuration, if we know the pressure everywhere thanks to CFD. It is convenient to part external and internal flows in such an analysis. External flows give contribution to drag including lift-induced drag; internal flows must follow Cp versus A (Area) curves not far from the monodimensional isentropic or polytropic flows in air intake. Three dimensional losses appear as a loss against optimum monodimensional flows.

So we get an exact evaluation of all the contributions to the propulsive loop if we get the pressure everywhere on the complete aircraft. Such a set of data is clearly the necessary base for final integration evaluation. It comes directly from CFD capability on complex configurations to rebuild the complete pressure distribution including the lips with the boundary layer region where measurements are scanty.

**Example**

As an example of such an approach, some results obtained from EULER and boundary layer computations, on Star vehicle (hypersonic aircraft) are presented on the following figures. Fig.1 presents the initial mesh before refinements. On Fig.2 is presented a cut into the symmetry plane, showing the different shocks. From the pressure distribution given by computations, we deduce the evolution of the equivalent Cp versus the area for the internal and external flow (see Fig.3). These curves, which the integral gives the total drag or thrust, show how the air intake integration has to be done with the best accuracy.

**Concluding remarks**

Integration of air intake by CFD is the only way to improve thrust minus drag and selecting of viable candidates to detailed experimental analysis in wind tunnel and after that in flight. From now an efficient methodology seems possible if a CFD capability of taking into account accurate complete geometry is available. It has been fixed that at present state of the art, progresses have to be done on two directions:—physical modelling improvement by carefull evaluation of experimental tests, done or to be done. For advanced workshops, a main emphasis is on carefull experimental test with better and more complete set of data.—first reference tests to improve or extend CFD capabilities to identify unsteadiness or transitional flows in air intake, in order to predict the boundaries of correct operation and the margins associated with the catastrophic degradation of the flow quality. Furthermore the flow unsteadiness has to be clearly evaluated at the edge of present CFD. However transition phase in the flow building around the configuration can be obtained by CFD and its validation is a major challenge at the present time. Unsteadiness evaluation and validation remain a target for the near future.

**References**

(1) P. Perrier "Concepts of Hypersonic Aircraft", Third Joint Europe/US, Short Course in Hypersonic RTW Aachen FRG 1990


Discrete models for the analysis of 2D wakes in unsteady aerodynamics

G. Riccardi, A. Iafraiti and R. Piva
Dip. Meccanica e Aeronautica
Università di Roma “La Sapienza”
Via Eudossiana, 18 Rome ITALY

Abstract A comparative numerical study of two-dimensional wake dynamics is presented. An application to the free wake motion behind an elliptical loaded wing on the Trefftz plane is considered first. The wake motion is approximated either by the dynamics of a set of vortices (vortex method) or by the dynamics of a piecewise linear curve (boundary element technique). In both approaches some considerations about the time integration accuracy control are made in terms of the most sensitive flow first integral: the Hamiltonian. Two types of the vorticity generation mechanism are tested in the second part of the paper where the application to the flow around a lentil at large incidence in an uniform stream is analyzed. Some aspects of the coupling between the vorticity production and the wake interaction are discussed in order to explain the periodic vorticity shedding.

1 Introduction

In the present paper we perform a comparative analysis between two different discretization techniques for the wake dynamics: the vortex method and the boundary element method. In particular we consider a free wake as well as the two wakes generated by a lentil in an uniform stream at large incidence. The latter case gives the opportunity to investigate the interaction mechanism between the wakes.

As a preliminary case we take into consideration the dynamics of a free wake. In particular we develop a numerical study for the motion of a vortex sheet behind an elliptically loaded wing on the Trefftz plane. A large emphasis is placed on the time integration accuracy control.

A study of the wake motion behind a lentil at large incidence is carried out in the second part of the paper. This case is complicated by the coupling with the vorticity production whose model is, in turn, strongly dependent on the wake dynamics. Typically the interaction between the upper and lower wakes leads to a periodic vorticity shedding. We have adopted two discrete approaches in order to study the kinematic of the flow field about the body, i.e. conformal mapping and boundary elements.

2 Free wake dynamics

The vorticity field of a 2D wake is given by

\[ \omega(z,t) = \int_0^{\Gamma(t)} \gamma(s) \delta(z - z(t,s)) \, ds \]

in which \( \gamma \) is the derivative with respect to the current natural parameter \( s \) of the circulation \( \Gamma(s) \) on each circuit around the wake arc \( C_i \), and \( \delta \) is the Dirac measure on \( \mathbb{R}^2 \). This vorticity field induces the velocity

\[ u(z,t) = \int_0^{\Gamma(t)} \gamma(s)K((z - z(t,s))) \, ds \]

where \( K = \nabla^2 G \) is the Biot-Savart kernel and \( G(z) = \frac{1}{2\pi} \log|z| \).

This field has a jump across the wake given by

\[ [u] = \gamma r \]

and the use of the tangential wake velocity definition

\[ w_r = \frac{1}{2}(u_r + u_t) \]

leads to the classical Birkhoff initial value problem for the wake motion [4]. Its numerical integration is carried out employing a preliminary kernel treatment that, avoiding the computation of a Cauchy integral, simplifies the overall discrete representation of the wake.

As a preliminary case we take into consideration the dynamics of a free wake. In particular we develop a numerical study for the motion of a vortex sheet behind an elliptically loaded wing on the Trefftz plane. A large emphasis is placed on the time integration accuracy control.

A study of the wake motion behind a lentil at large incidence is carried out in the second part of the paper. This case is complicated by the coupling with the vorticity production whose model is, in turn, strongly dependent on the wake dynamics. Typically the interaction between the upper and lower wakes leads to a periodic vorticity shedding. We have adopted two discrete approaches in order to study the kinematic of the flow field about the body, i.e. conformal mapping and boundary elements.

In the following we indicate by \( K_\epsilon \) both the desingularized and the regularized kernel, where \( \rho \) identifies with \( \epsilon \) and \( \delta \) respectively.

The wake is approximated with a set of vortices (Vortex Blobs method) or with a straight panel representation (Boundary Elements technique). In the former case we study the vortex dynamics by performing an approximate integration of the following ordinary differential system of \( N_{VB} \) equations

\[ \begin{align*}
\frac{dz_{PB}^i}{dt} &= \sum_{j=1}^{N_{VB}} K_\epsilon^2(z_{PB}^j - z_{PB}^i) \\
z_{PB}^i(0) &= z_0^i(s_0) 
\end{align*} \]
where \( \Gamma \) is the circulation around the \( j \) -th vortex. In the second approach we study the evolution of the piecewise linear curve approximating the wake by following the motion of its vertices. This leads to the system of \( N_{BE} \) equations

\[
\begin{align*}
\frac{dx_{t}^{BE}}{dt} &= \sum_{i=1}^{N_{BE}} \int_{s_{i}}^{s_{i+1}} K_{s}^{1}(x_{t}^{BE} - z_{j}^{BE}(t, \sigma_{t})) \, d\sigma_{t} \\
x_{0}^{BE}(0) &= z_{0}(\sigma_{0})
\end{align*}
\]

where \( \gamma_{t} \) is assumed constant along any panel.

A direct evaluation of the numerical time-integration accuracy in both VB and BE discrete approaches is possible by comparing suitable quantities related at some flow first integrals. From our numerical experience we deduce that the most sensitive flow invariant is the Hamiltonian defined by

\[
\mathcal{H}_{e}^{BE}(x) = G(x) + \frac{1}{4\pi} \log \left( \frac{|x|^{2} + \epsilon^{2}}{|x|^{2}} \right)
\]

in which for the two above choices of \( K_{s}^{1} \) the smoothed Green’s functions are

\[
G_{s}^{BE}(x) = G(x) + \frac{1}{4\pi} \sum_{k=1}^{K_{s}} a_{k} E_{s}(c_{k} |x|^{2})
\]

where the smoothed Green’s function is also a first integral for the discrete vortex model whereas \( \mathcal{H}_{e}^{BE} \) is not a stationary quantity. In fact the flow maps a set of points in a set of points but it does not transform a set of straight segments in a set of straight segments. Hence

\[
\frac{d\mathcal{H}_{e}^{BE} \, x_{t}^{BE}}{dt} = 2 \sum_{k=1}^{N_{BE}} \mathcal{H}_{e}^{BE} \left( \int_{s_{i}}^{s_{i+1}} d\sigma_{t} \right) \left( z_{j}^{BE}(t, \sigma_{t}), t \right) \cdot \frac{dx_{t}^{BE}(t, \sigma_{t})}{dt} \neq 0
\]

where \( u_{j} \) is the approximate self-induced velocity field, while \( \frac{dx_{t}^{BE}}{dt} \) is its linear interpolation given by the BE description of the wake motion. The numerical integration of the above equation allows for a control of consistency between the wake velocity field and its displacement by comparing with the numerical values obtained by the corresponding discretized form of (2).

We apply the previous techniques to the numerical simulation of a wake shedded from an elliptically loaded wing on the \( Trefftz \) plane (see fig.1). As already described in \( \phi \) where a VB approach with a desingularization of the Biot-Savart kernel was used, both discrete approaches for a fixed smoothing parameter \( \rho \) lead to a converged numerical solution of equation (1) with respect to the refinement of both the initial space discretization and the time integration step. Then it is possible to find for any \( \rho \) a well defined limit solution, but unfortunately the resulting sequence for \( \rho \to 0^{+} \) does not converge.

However, as shown in \( \phi \), the loss of convergence appears confined in a smaller and smaller neighbourhood of the sheet tips. These calculations show that the time integration accuracy control requires a very large computational effort for the BE approach, whereas it results simple and fast for the VB formulation. In fact the VB approach appears faster than the BE formulation even if the time integration accuracy control is neglected.

3 Wake dynamics past a lentil

The above numerical techniques are applied to the simulation of the flow around a lentil placed in an uniform stream at large incidence. To this aim we have to introduce models for the velocity field about the body and for the vorticity production at the sharp edges where the separation occurs.

Velocity field representation

The first task is reached by following two different paths: the conformal mapping analysis [9] and the Poincaré representation [2].

In the former way the field external to the body (z plane) is mapped in the exterior of a unit circle (s plane) by the Karman-Trefftz transformation. The wakes are discretized with a suitable set of vertices (to be considered points out of the flow field) and the motion of the \( k \) th vortex is obtained by a numerical integration of the equation

\[
\frac{dz_{k}}{dt} = \frac{d}{dz} \left( W - \frac{\Gamma_{k}}{2\pi i \xi - i\xi} \right) \left( dz_{l} \right) |_{l} + \frac{1}{2\pi i} \left( \frac{dz_{k}}{dz} \right) |_{l}
\]

together with the smoothing procedure [9] for the vortex interactions.

Using the Poincaré representation we decompose the velocity field in the sum of a uniform translational flow \( u_{0} \) plus a perturbation field \( u \) generated by the presence of the body. The limit of this field on the body is obtained by solving the integral equation.
which starts from finite velocity one (used only in the conformal approach) the nascent vortices two different schemes for the production of vorticity. In the first incidence of the lentil production of vorticity induced one is difficult. We observe this mechanism only for the discretized either. In the computational procedure the two issuing wakes may be and delays the new formation of the lower wake. Moreover their

\[ 1 \leq w(r) - \int_{\partial \Omega_0} w_0(\eta) K(\xi - \eta) \cdot r(\xi) d\alpha(\eta) = \frac{1}{2} \int_{\partial \Omega_0} w_0 \cdot r(\xi) d\alpha(\eta) + \sum_{n=1}^{2} r(\xi) \cdot \int_{\partial \omega_n(t)} \gamma_n^T(\xi) K(\xi - Z_n(t, \xi)) d\alpha(\eta), \]

where \((r, w)\) is the intrinsic reference system on the body boundary \((\partial \Omega_0)\) and \(\omega_n(t)\) is the configuration of the wake 1 and 2 respectively. In the computational procedure the two issuing wakes may be discretized either by VB or by BB technique.

Production of vorticity

When a vortex discretization of the wake is adopted, we consider two different schemes for the production of vorticity. In the first one (used only in the conformal approach) the nascent vortices are placed on the symmetry plane of the body at a small distance from the corresponding edge and their intensity is fixed by imposing a finite velocity at the lentil edges. In the second one (adopted also in the Poincaré formulation) we follow the motion of a neutral particle which starts from the body edges with velocity

\[ w = \frac{1}{2} (u(P^+) + u(P^-)) \]

(where \(P^+, P^-\) are points near the edge on opposite sides of the wake) during a suitable time interval. Hence we assign a circulation around the particle obtained by integrating the vorticity generated in that time interval. A similar approach is adopted also with a panel discretization of the wake, but the amount of shedded vorticity is scattered along the nascent panel.

In the present numerical simulations we assume the incidence of the lentil sufficiently large to have two separation points located on the body edges. Two wakes are produced behind the body and their interaction plays an essential role in the flow field development: the main effect is a periodic wake detachment that leads to the formation of the typical vortex street shown in fig.2.

The comparison between the conformal mapping and the Poincaré description of the velocity field gives a good agreement with respect to certain global flow quantities such as the body circulation \(\Gamma\) or the time history of \(\frac{d\Gamma}{dt}\) even by assuming different vorticity production mechanisms. From this comparative analysis the method of the neutral particles appears more sensitive to the velocity perturbations near the edges than the finite velocity approach. Moreover the comparison between the two wake discretizations has shown an equal mean behaviour with some superposed oscillations that appear more persistent for the BB technique, owing to the wake continuity constraint.

The flow resulting from an impulsive starting evolves towards a periodical one after a short initial transient. This periodicity is forced by the process of interaction between the wakes, shown in fig.3, that can be summarized as follows. Alternatively one wake (say 1) rolls up, and leaves the generating edge inducing the convection of the other wake (say 2) into the region between the first one and the body. The motion of the wake 2 leads to a strong local stretching of the wake 1 up to its breaking in a region close to the edge. When the velocity field induced by the lower wake (see fig.3b) is sufficiently high to detach one or more elements from the upper wake the above phenomenon is accompanied by the convection of these elements up to the lower edge. This kind of interaction leads to the formation of several small clusters in which the elements coming from the upper wake couple with some elements of the lower wake. The presence of these clusters in the neighbourhood of the edge influences the local velocity field and delays the new formation of the lower wake. Moreover their effect on the vorticity production mechanism enhances the oscillatory behaviour of \(\frac{d\Gamma}{dt}\) even if a separation of this effect from the self-induced one is difficult. We observe this mechanism only for the lower edge, while it may occur also at the upper edge only for large incidence of the lentil.

In order to reduce the CPU time we adopt a coalescence procedure of the far wake regions replacing a cluster of wake elements with a single vortex. At the present time the effects of this approximation are not completely understood: it seems that this causes a little reduction of the circulation released and a short delay in the time history of the circulation around the body. Also the choice of the coalescence parameters (and obviously the choice of the procedure itself) is very critical because we have experimented that this can influence the vorticity production in a very subtle way.

![Fig.2 a) Vortex street behind a lentil at incidence 60°. b) Circulation shed at the leading-edge (dotted line), trailing-edge (solid line) and global circulation around the wakes (dash-dot line)](image)
Finally carrying out several other calculations with different incidences it appears a monotonic decrease of the vorticity shedding frequency for growing incidence which obviously vanishes when the lentil is normal to \( \omega_0 \). We have also analyzed the flow field for different values of the lentil thickness at a fixed incidence of \( 60^\circ \): the shedding frequency decreases monotonically with the lentil thickness up to reaching the flat plate frequency. This behaviour appears essentially due to the increasing distance between the wakes that slows down their interaction.

References


INITIAL TEMPERATURE FIELD FOR UNSTEADY LAMINAR
FORCED CONVECTION FROM AN IMPULSIVELY STARTED SPHERE

Lai-Chen Chien
Institute of Physics, Academia Sinica
Taipei, Taiwan 11529
Republic of China

Abstract—Analytic solution for forced convection heat transfer from an impulsively started heated sphere is investigated. Because of the impulsive start, there is a singularity at the very beginning of the motion. The accurate analytic solution for the initial temperature field is obtained by solving the non-linear energy equation using the method of matched asymptotic expansion to the third order. The solution is in terms of exponential and error function. The time development of the temperature field is plotted and investigated. The local Nusselt number over the sphere surface and the progress of minimum Nusselt number point with time are obtained.

I. INTRODUCTION

The problem of incompressible viscous flow over an impulsively started sphere has been studied by many investigators (Bentwich and Miloh, 1978). The extension of the problem to the heat transfer has also been the subject of numerous analytical and experimental investigations (Chen and Mucoglu, 1977). Several theoretical investigations have been reported centering around the classical problems of heat and mass transfer from a solid sphere into a low Reynolds number velocity field. Sano (1981), Bentwich and Miloh (1978) obtained the asymptotic solutions for Stokes-Oseen flow using the method of matched asymptotic expansions. Aris and Taylor (1962) used singular perturbation technique expanded the Nusselt number in terms of Peclet number, and yielded an accurate expression for the rate of transfer of energy for small Reynolds number. Hieber and Gebhart (1969) studied mixed convection from a heated sphere for Stokes flow. Chen and Mucoglu (1977) solved the conservation equations of the non-similar boundary layer using finite difference method to study the combination forced and free convection about a sphere for small Reynolds number.

In this study, we try to employ the inner-outer expansion method to solve the axi-symmetrical unsteady Navier-Stokes and energy equations in an attempt to extend the boundary layer theory to larger Reynolds number. The stream function for the initial flow field is obtained by the matched asymptotic expansion to the third order (Chien and Chen, 1984) is used to solve the energy equation. The analytic solution to the third order for the temperature distribution corresponding to the initial flow field is treated with great care because of the complicated mathematical operation. The temperature field is thus obtained in terms of exponential and error function. The time development of the temperature field properties such as the local Nusselt number over the sphere surface, and the progress of minimum Nusselt number point with time is investigated.

II. BASIC EQUATIONS

Consider a solid sphere of radius r0 which is started impulsively from rest and subsequently moves with constant velocity U. The origin of the spherical coordinate system is fixed at the center of the sphere with the axis θ = 0 in the direction opposite to the motion of the sphere. Assume that the fluid is an incompressible viscous continuum with constant properties, and that the effect of the heating of the fluid by viscous dissipation from sphere surface is neglected. With the foregoing assumption, the Navier-Stokes equation for the fluid motion can be expressed in the form (Chien and Chen, 1984) as

\[ \frac{\partial \psi}{\partial t} + \frac{\partial}{\partial r} [r \frac{\partial \psi}{\partial r}] + \frac{\partial}{\partial \theta} (r \sin \theta \frac{\partial \psi}{\partial \theta}) + \frac{\partial}{\partial z} \frac{\partial \psi}{\partial z} = 0 \]

where \( \psi \) is non-dimensionalized by the characteristic time \( t_0 \), radial coordinate \( r \) by the radius of the sphere \( r_0 \), stream function by \( r_0 U \). The parameters are defined by \( c = \frac{U_0}{r_0} \), \( \alpha = \frac{1}{(2 \text{Re})} \), where Reynolds number is based on radius. For the problem of initial flow, \( t_0 \) is a small quantity much less than 1 and \( \alpha \) is order 1. The boundary conditions are non-slip at the surface of the sphere, and the uniform free stream conditions far from the sphere.

The flow field is divided into two regions. One is the inner region close to the sphere surface and the outer is the outer inviscid region, i.e. \( \alpha = 0 \). The outer solution to the third order for the stream function is obtained by Wang (1969). The inner solution to the third order by the method of inner-outer expansion is given by Chien and Chen (1984). The composition stream function is

\[ \psi = \frac{1}{2} \left( \frac{1}{r^2} - \frac{1}{2} \right) \sin^2 \theta + 3 \int_0^t \frac{1}{2} \left( \frac{1}{2} - \frac{1}{4} \right) \cos \theta d\theta \]

The boundary conditions are non-slip at the surface of the sphere, and the uniform free stream conditions far from the sphere.

The flow field is divided into two regions. One is the inner region close to the sphere surface and the outer is the outer inviscid region, i.e. \( \alpha = 0 \). The outer solution to the third order for the stream function is obtained by Wang (1969). The inner solution to the third order by the method of inner-outer expansion is given by Chien and Chen (1984). The composition stream function is

\[ \psi = \frac{1}{2} \left( \frac{1}{r^2} - \frac{1}{2} \right) \sin^2 \theta + 3 \int_0^t \frac{1}{2} \left( \frac{1}{2} - \frac{1}{4} \right) \cos \theta d\theta \]

The boundary conditions are non-slip at the surface of the sphere, and the uniform free stream conditions far from the sphere.

The flow field is divided into two regions. One is the inner region close to the sphere surface and the outer is the outer inviscid region, i.e. \( \alpha = 0 \). The outer solution to the third order for the stream function is obtained by Wang (1969). The inner solution to the third order by the method of inner-outer expansion is given by Chien and Chen (1984). The composition stream function is

\[ \psi = \frac{1}{2} \left( \frac{1}{r^2} - \frac{1}{2} \right) \sin^2 \theta + 3 \int_0^t \frac{1}{2} \left( \frac{1}{2} - \frac{1}{4} \right) \cos \theta d\theta \]

The boundary conditions are non-slip at the surface of the sphere, and the uniform free stream conditions far from the sphere.

The flow field is divided into two regions. One is the inner region close to the sphere surface and the outer is the outer inviscid region, i.e. \( \alpha = 0 \). The outer solution to the third order for the stream function is obtained by Wang (1969). The inner solution to the third order by the method of inner-outer expansion is given by Chien and Chen (1984). The composition stream function is

\[ \psi = \frac{1}{2} \left( \frac{1}{r^2} - \frac{1}{2} \right) \sin^2 \theta + 3 \int_0^t \frac{1}{2} \left( \frac{1}{2} - \frac{1}{4} \right) \cos \theta d\theta \]

The boundary conditions are non-slip at the surface of the sphere, and the uniform free stream conditions far from the sphere.

The flow field is divided into two regions. One is the inner region close to the sphere surface and the outer is the outer inviscid region, i.e. \( \alpha = 0 \). The outer solution to the third order for the stream function is obtained by Wang (1969). The inner solution to the third order by the method of inner-outer expansion is given by Chien and Chen (1984). The composition stream function is

\[ \psi = \frac{1}{2} \left( \frac{1}{r^2} - \frac{1}{2} \right) \sin^2 \theta + 3 \int_0^t \frac{1}{2} \left( \frac{1}{2} - \frac{1}{4} \right) \cos \theta d\theta \]

The boundary conditions are non-slip at the surface of the sphere, and the uniform free stream conditions far from the sphere.
The homogeneous part of equation (11) is of the form

\[ \frac{1}{\nu^2} \left( \frac{\partial^2 T}{\partial \xi^2} \right) = \frac{4}{\sqrt{\nu}} e^{-\frac{\nu}{\sqrt{\nu}}} \]

where \( f_1 \), \( f_3 \), \( F_3 \) and \( g_3 \) are polynomials of the similar forms.

In order to solve the energy equation to obtain the temperature field for the convection problem considered, the equation is written as (Chien and Kung, 1982)

\[ \frac{3}{r^2} \sin \theta \frac{\partial}{\partial r} \left( r^2 \frac{\partial T}{\partial r} \right) + \frac{1}{r^2} \frac{\partial}{\partial \theta} \left( r^2 \frac{\partial T}{\partial \theta} \right) + \frac{\partial^2 T}{\partial z^2} = 0 \]

where dimensionless temperature \( T \) is \((T' - T_{m})/(T_{w} - T_{m})\), \( T' \) is dimensional temperature, and \( T_{w} \), \( T_{m} \) are the temperature on the sphere surface and at infinite respectively. When the sphere is impulsively started to move with a constant velocity, a constant temperature difference between the solid sphere and the fluid is suddenly imposed. The initial and boundary conditions are

\[ T = 0 \quad \text{for} \quad r < 0 \]  
\[ T = 1 \quad \text{at} \quad r = 1 \]  
\[ T \rightarrow 0 \quad \text{as} \quad r \rightarrow 0 \quad \text{for} \quad r > 0 \]

III. ANALYTIC SOLUTION FOR THE TEMPERATURE FIELD

Similar to the problem of convective heat transfer over an impulsively started circular cylinder (Chien and Kung, 1982), we construct the solution for the temperature field by the method of additive composition. Sano (1978) has shown that the outer solution for the energy equation vanishes concerning the heat transfer for the flow over an impulsively started cylinder. The conclusion is also valid for the sphere problem considered in this investigation.

It can be shown that the inner solution is of the form

\[ T^i = T_1(R, \theta, t) + cT_2(R, \theta, t) + c^2T_3(R, \theta, t) \]

where \( R \) is the stretched radial coordinate and defined by \( R = (R - 1)/c \). Substituting the above expression into (3) and collecting the terms of the same order \( c \), we have the equations

\[ T_{1c} - (a/Pr) T_{1RR} = 0 \]  
\[ T_{2c} - (a/Pr) T_{2RR} = (\phi T_{1RR} + c^4 T_{1R} \sin \theta) \]  
\[ T_{3c} - (a/Pr) T_{3RR} = (\theta T_{2RR} + c^4 T_{2R} \sin \theta) \]

for the first, the second and the third order respectively.

And the boundary conditions for the above equations are

\[ T(R, \theta, t) = 1 \quad \text{at} \quad R = 0 \]  
\[ T_1(R, \theta, t) = 0 \quad \text{at} \quad R = 0 \quad \text{for} \quad n = 2, 3 \]  
\[ T(R, \theta, t) = 0 \quad \text{at} \quad R = \infty \quad \text{for} \quad n = 1, 2, 3 \]  

The solution for the first order is similar to that obtained by Sano (1978) solving the convection of cylinder problem,

\[ T_{1} = \text{erf} \mu \]

where \( \mu \) is \( \sqrt{\frac{1}{\nu}} n \) and \( \eta \) is the inner stretched variable \( \eta = R/(\sqrt{\nu} \theta) \). Substituting the first order solution (9) and the inner stream function into the right hand side of the second order equation (6), one has

\[ T_{2c} - aT_{2RR}/Pr = -6Pr[(n-1)/\sqrt{\nu}] e^{-\eta^2} \text{erf} \eta + e^{-\xi^2}/\sqrt{\nu} \]

where \( \xi = \sqrt{Pr} n \mu \). Assume the solution for \( T \) be of the form \( T_2 = \text{erf}(\mu \cos \theta) \). Then the equation (10) becomes

\[ T_2''(\eta) + 2PrT_2'' = 4Pr[(n-1)/\sqrt{\nu}] e^{-\eta^2} \text{erf} \eta + e^{-\xi^2}/\sqrt{\nu} \]

The homogeneous part of equation (11) is of the form

\[ T_2''(\eta) + \eta T_2'' = 0 \]

The solution may be in terms of error function and Hermite polynomial (Courant and Hilbert, 1953, p.90). The particular integral for (11) is obtained by the method of undetermined coefficients. Then we have the complete solution

\[ T_2(\eta) = \left [ \text{erf}(\mu \cos \theta) + 4Pr[(n-1)/\sqrt{\nu}] e^{-\eta^2} \text{erf} \eta + e^{-\xi^2}/\sqrt{\nu} \right ]/8 \]

where \( C_2 = 12(Pr+1)^2Pr/[1(3Pr+1)] \) and \( \eta = \text{erf} \mu \theta \).

The right hand side of equation (7) is function of derivatives of the first and the second order expansion. Substituting (9) and (13) into the right hand side of equation (7), we have

\[ T_{3c} - aT_{3RR}/Pr = [(M_{31} \eta + M_{32} \eta) \cos \theta] \]

where coefficients of the triangular functions are polynomial of error and exponential function. We assume the solution of Equation (14) be of the form

\[ T_3 = 3\sqrt{\nu} Pr/\pi F_{31}(\eta) + F_{32}(\eta) \cos \theta + \left [ F_{33}(\eta)/2 \right ] c^2 \sin^2 \theta \]

Substituting the above expression into equation (14), it gives set of six equations in the form

\[ F''(n)+2PrF''-2nPrF \]

\[ = \text{erf}(\mu \cos \theta), \exp(-n^2), \exp(-n^2) \]

642
The comletementary solution of (16) is able to be expressed in terms of error function and Hermite polynomial. And the particular integrals are obtained by the method of undetermined coefficients. The complicated mathematical operation is repeated with great care.

To sum up, the solution of the energy equation is

\[
T = e^{\alpha t} F_2(n) \cos \theta + \left[ F_3(n) + 3\sqrt{Pr/(\pi Re)} F_4(n) \right] \cos \theta + \frac{\pi \sin \theta}{2} \left[ -9 Pr/F_3(n) + F_5(n) / 2 \right] \sin \theta.
\]

where \( F_2(n), F_3 \) to \( F_5 \) are polynomial of error function and exponential function.

IV. RESULTS

The temperature field (17) is plotted in Figure 1 for \( Re = 100, Pr = 0.7 \). At short time after the impulsively start, the isotherms in the front half part are almost parallel to the surface of the sphere. As time goes on, at \( \tau = 0.6 \), the isotherms displaced away from the sphere as the viscous layer thickens. Similar phenomenon is shown in Figure 2 for \( Re = 500, Pr = 0.7 \) at \( \tau = 0.6 \). The higher the Reynolds number, the thinner the thermal boundary layer because the convection becomes more effective in displacing the streamlines downstream away from the sphere.

Having computed the temperature field, we can estimate the heat transfer between the sphere and the surrounding fluid.

\[
Nu = -\frac{K(\alpha t/3r)}{\tau = 1}.
\]

The local Nusselt number distribution around the surface of the sphere for \( Pr = 0.7 \) at \( Re = 100 \) and 500 is shown in Figure 3. Because no existing investigation is available for comparison, we compare the trends of this solution with those obtained in studying cylinder by Sano (1978), Chien and Kung (1982). The similar results are obtained.

We differentiate the Nusselt number distribution function, set it equal to zero, and find the minimum Nusselt number point. By Newton's method, we can obtain the progress of minimum Nusselt number point at the sphere surface with respect to time, Figure 4.

V. CONCLUSIONS AND RECOMMENDATIONS

The major objective of the present investigation has been to work out the short time solution for unsteady forced convection heat transfer from an impulsively started sphere. Because of the impulsive start, there is a singularity behavior of the flow field at the very beginning of the motion. An accurate solution is obtained by the method of asymptotic expansion to the third order. The expansion is valid for the short time only, \( \tau < 1 \). The viscous layer considered flow field (Chien and Chen, 1984) and temperature field obtained in this investigation can be used as initial conditions for numerical computation. And the solution can be continued by numerical integration to obtain the larger time solution (Schlichting, 1979, p. 149). Using the accurate stable numerical method to solve the governing equations, we can depict the time history of the flow patterns and the temperature fields.

REFERENCES


Figure 1. Isothermal for Pr=0.7 at Re=100, Time=0.6.
Figure 2. Isothermal for Pr=0.7 at Re=500, Time=0.6.

Figure 3(a). Nusselt number distribution on sphere surface for Pr=0.7 at Re=100.

Figure 3(b). Nusselt number distribution on sphere surface for Pr=0.7 at Re=500.

Figure 4. Progression of minimum Nusselt number point with time.
MARKOV A.A.
Institute for Problems in Mechanics
USSR Academy of Sciences
Prospect Vernadskogo 101
117526 Moscow, USSR.

Abstract- The analysis and generalization of known algebraic models of turbulence have been carried out for three-dimensional boundary layer computations the supersonic flow over finite, twisted wings, taking into account the effects of aerodynamic heating and heat radiation. The results of heat-transfer prediction are in good agreement with experimental data.

Supersonic laminar flows at moderate Reynolds numbers have been simulated on the basis simplified Navier-Stokes equations by space-marching method, combined with global pressure iterations. The generalization of known parabolization technique for equations in subsonic regions of the flow and new algorithms of relaxation of the pressure are suggested. The results of three-dimensional shock layer computations over blunt bodies are presented.

I. INTRODUCTION

In recent papers on numerical computation of viscous gas flows it has been shown that the calculation efficiency may be enhanced by using the boundary-layer methods and appropriate scales both in complete Navier-Stokes (N.-S.) equation and simplified N.-S. composite asymptotic equations, namely viscous layer (VL) and viscous layer with azimuth diffusion (VLAD) equations.

The application of boundary layer (BL), VL and VLAD equations and space-marching algorithm along the dominant direction of the flow, combined with the global pressure iteration allows a tremendous reduction in necessary computing time and storage requirements over that required for time-dependent approach for N.-S. equations.

II. LAMINAR FLOWS

We shall write the simplified Navier-Stokes equations in arbitrary curvilinear coordinate system in strictly conservative form.

Let \( x^1, x^2, x^3 \) be Cartesian coordinates; \( x^i, x^j, x^k \) - curvilinear coordinates of a point \( M \). Let us introduce covariant \( \bar{a} \) and contravariant \( a^i \) basises:

\[
\bar{a}^i = \left( \frac{\partial}{\partial x^1}, \frac{\partial}{\partial x^2}, \frac{\partial}{\partial x^3} \right), \quad a_i = \left( \partial x^1 / \partial \bar{x}^i, \partial x^2 / \partial \bar{x}^i, \partial x^3 / \partial \bar{x}^i \right)
\]

then components \( g_{ij}^1, g^{ij} \) of metric tensor can be expressed as follows:

\[
g_{ij}^{1} = \bar{a}_{i} \cdot \bar{a}_{j}; \quad g^{ij} = \bar{a}^{i} \cdot \bar{a}^{j}
\]

This work supported by a grant of Soviet-French Italian joint venture in computer and software business 'INTERQUADRO'.

RIZHOV YU.A., SCHECKIN G.A.
Moscow Aviation Institute
Volokolamskoe highway 4
125871 Moscow, USSR.

Let us denote \( g = \det \left| g_{ij} \right| \) and consider vectors \( \tilde{a}^1, \tilde{a}^2, \tilde{a}^3 \) at two points \( M \) and \( M_0 \), employing index zero for point \( M_0 \).

We shall introduce the coefficients \( G^j_k \)

\[
G^j_k(M_0, M) = \tilde{a}^j_0 \cdot \tilde{a}_k, \quad \text{then} \quad G^j_k(M_0, M) = \delta^j_k
\]

We shall use operators \( L_{VL} \) and \( L_{VLAD} \) for

\[
L_{VL}(f) = 0, \quad L_{VLAD}(f) = 0, \quad \text{where} \quad f = (u^1, u^2, u^3, p, h)^T, \quad \text{velocity vector} \quad V = u^1 \tilde{a}^1 + u^2 \tilde{a}^2 + u^3 \tilde{a}^3 = u^1 \tilde{a}^1 = u^1 \tilde{a}^1 . \quad \text{The summation on repeated indexes is usually assumed, if exception is not noted.}
\]

\[
L(f) = \frac{\partial}{\partial x^1} \left( A^1(f) - \frac{1}{Re} \frac{\partial}{\partial x^1} \left( \frac{\partial g^{ij}}{\partial x^1} + g^{ij} \right) \right)
\]

Here \( j = s = 3 \) for \( VL \) and \( j_s = 1, 2, 3 \); \( s_j = j + 1 \) for \( VLAD \) equations. Components of vectors \( A^1 = (A^1_1, \ldots, A^1_5) \) and matrices \( \left| A^1 \right| \), \( \left| \varepsilon \right| \), \( \left| \varepsilon \varepsilon \right| \) for the point \( M_0 \) can be written as follows:

\[
(\varepsilon)_{10} = \sqrt{g} g_{ij}^{1 \varepsilon} \mu^{j1} u^{i1} + \mu^{i1} u^{i1} \varepsilon^{11} (\varepsilon_{11} - \delta^{11})
\]

\[
(\varepsilon)_{14} = (\varepsilon)_{14} = (\varepsilon)_{14} = 0, 1, m = 1, 2, 3, \quad \varepsilon_{14} = \frac{(Y - 1) \rho}{Y - 1}
\]

\[
(\varepsilon)_{10} = g_{ij}^{1 \varepsilon} \mu^{j1} u^{i1} (\varepsilon_{11} - \delta^{11}) + \mu^{11} (\varepsilon_{11} - \delta^{11}) + \varepsilon_{10} \varepsilon_{10}, \quad (\varepsilon)_{10} = 0, \quad (\varepsilon)_{10} = \sqrt{g/\rho} g_{ij}^{1 \varepsilon} ;
\]

\[
\varepsilon_1 = \sqrt{g} g_{ij}^{1 \varepsilon} u^{i1} u^{j1} + \mu^{i1} u^{i1} \mu^{j1} u^{j1} + \mu^{i1} u^{i1} \mu^{j1} u^{j1}
\]

\[
(\varepsilon)_{10} = 0, \quad (\varepsilon)_{10} = \sqrt{g} (g_{ij}^{1 \varepsilon} u^{i1} \frac{\delta^{11}}{\partial x^1} + \mu^{i1} u^{i1} \frac{\delta^{11}}{\partial x^1})
\]

here \( \tilde{\nu} \) is second viscosity coefficient.

We use approximate formulae for components \( A^1_1, A^1_5 \) of vector \( A^1 \) as follows:

\[
A^1_1 = \sqrt{g} (\mu u^1)^2 + \chi_1 \rho + \chi_1 (1 - \chi_1) P \quad (1)
\]

\[
A^1_5 = \sqrt{g} (\mu) (u^2)^2 + \frac{\chi_2}{\chi_2} u^1 (1 - \chi_2) P, \quad (2)
\]

where \( \chi_1, \chi_2 \) are special functions discussed below and \( P \) is a part of pressure, that is assumed to be known from previous global iteration. If \( \chi_1 = \chi_2 = 1 \), or \( P = 0 \), than formulæ become exact. For stability of space-marching algorithm is sufficient, that \( \bar{u} > 0 \) and \( \bar{u} \bar{u} = \omega_1 \bar{u} (M_1, M_2, M_3) \); \( \omega < 1, \chi_2 > 0 \).
Here $F_1=1$, if $M_1 > 1 + c$ and

$$\gamma R_1^2 + (\gamma-1) b_1 (M_1, M_2), \quad \text{if } M_1 \leq 1 + c,$$

$\chi$: (M_1, M_2) $\gamma$ and $\gamma$: (M_1, M_2) $\chi$.

The azimuth skin friction coefficient distributions $Re$ converge to $sufficient at average N$ for global iterations of number $R(x)$.

$V_2x/(l+x)$ at swept angle $\alpha$.

$\delta$: layer thickness are in good agreement with known data. $\times$ cones and swept infinite wings. The results than $Q$ at section $X$.

$\delta$ and $\delta$ are obtained $\times$ conic and bielliptic cone $\delta$. and dots are presented $CR$ and $CF$ at $x$.

$\delta$ depends on kinematic viscosity coefficient $\nu_1$. The value of $\nu_1$ varies along the wing surface, but this variation is not large so approximation $k_0$: const has been used.

The computations were made using turbulence models by Fletcher and Cebeci, including corrections. The results are given at attack angle of $15$ for section $X = 0.33, 0.6, 0.9$ (where $x = x/c$, $c$ is local wing chord). These sections correspond to region of fully-developed turbulence flow.

The computed $q$ at section $X = 0.33$ is approximately on $5$ per cent over and at $X = 0.9$ on $5$ per cent lower the corresponding values of $Q$.

The computed values of $q$ for Cebeci model without correction is on $7$ per cent lower, than $Q$ at section $X = 0.33$, but at sections $X = 0.6, X = 0.9$ the values $q$ and $Q$ are almost the same. The addition Klebanoff and Rotta corrections to Cebeci model has not changed essentially the values of $q$. The use Cebeci model with correction of local turbulent Prandtl number gives increasing of $q$ on $2 - 4$ per cent. The results of $q$ computation, using Spalding model are approximately on $200$ per cent over, than $Q$.

The analysis of computed results allows us to modify Cebeci model of turbulence. Instead of Clauser formula for outer vorticity layer the following approximation is suggested

$$\mu_1 = k_0 + v_1 \delta^2,$$

where $\delta - \delta$ is boundary layer displacement thickness, $k_0$ is dimensional constant, that depends on kinematic viscosity coefficient $\nu_1$. The value of $\nu_1$ varies along the wing surface, but this variation is not large so approximation $k_0$: const has been used.

The computations on windward side, slender plane delta wing $65$-sweep, sharp leading edges have been carried out for BL equations, including laminar, turbulence transition and turbulent regions of flow. The solution of Euler equations has been obtained by Godunov method.

Effects of turbulence are incorporated by specifying viscosity coefficient according to three turbulence closure models by Spalding, Fletcher and Cebeci with some corrections, that have been found in accordance with experimental data for heat transfer coefficient. The anisotropy of eddy viscosity investigation for Rotta model has been developed, using Klebanoff correction for intermittency at outer vorticity layer and correction for turbulent Prandtl number variation across the boundary layer.

The heat flux distribution over delta wing surface is compared with experimental data $Q$ in Fig. 3. The windward side heat transfer distributions are shown for section $\delta = 0.25$ ($\delta = z/l$; $l$ is semi-span wing size).

Experimental data have been obtained in TzAGI.

Fig. 1.

Fig. 2.

Fig. 3.
THE OPTIMUM SHARE OF HYDROFOIL BENEATH A FREE SURFACE

A.H. Essawy and A.Y. Al-Hawaj

Abstract

The usual assumptions in problems of the obstacle beneath a free surface are taken as a basis: namely, the liquid is non-viscous and moving two-dimensionally steadily and without voracity, the only force acting on it is gravity. With these assumptions together with a linearization assumption we determine the forces, due to the hydrofoil to obtain the optimum shape so that the drag is minimum.

I. Introduction:

There is an extensive literature connected with wave resistance due to a submerged obstacle (see, e.g., Havelock, T.H. (1), (2), Kochin, N.E. (3), Kothcin, N.E. (4), Wehasen, J.V. and K. Kaitun, E.V. (7), Kreisel, G. (5) and Ribouchinsky, D. (6) but the three papers which are most relevant to present work are those to Kochin, N.E.; Kibeil, I.A. and Roze, N.V. (3) and Wu, T.Y. and Whitney, A.K. (8) and Essawy, A.H. (9)

A singular integral equation of the boundary value problem is obtained and can be solved to yield expressions for the lift and drag as functions of the unknown singularity distribution, of being the vortex strength together with the known shape, z (hydrofoil slope), which yields:

A singular integral equation of the boundary value problem is obtained and can be solved to yield expressions for the lift and drag as an admissible function which minimize

II. The Hydrofoil Beneath a Free Surface

A hydrofoil of arbitrary shape is in steady, rectilinear motion at a depth h beneath the free surface of a uniform liquid flow with speed \( U \) in the x-positive direction. We assume the liquid is non-viscous and moving two-dimensionally and without voricity, the only force acting on it is gravity.

The problem will be solved on the basis of linearized theory and for this purpose we introduce the following vortex distribution on the x-axis:

Vortices of strength \( \gamma(z) \) per unit length in \( 0<\xi<r, y=0, (\gamma>0, \text{clockwise}) \)

The complex potential due to a single vortex at \((\xi, h)\) as follows:

\[
\psi(z) = \frac{\xi}{2\pi i} \log \frac{z-\xi}{\xi} + \frac{\gamma}{2\pi} \int \frac{dz'}{z-z'} \cdot \text{for } y=0, \text{c=xi-h).} \tag{2.1}
\]

The complex potential due to the complete distribution of vortices as described above will be:

\[
\psi(z) = \frac{\gamma(z)}{2\pi i} \log \frac{z-c}{z-\xi} + \int \frac{dz'}{z-z'} \cdot \text{for } y=0, \text{c=xi-h).} \tag{2.2}
\]

Denoting the potential of this steady motion by

\( \Phi = U + i \Psi \tag{2.3} \)

Let the x- and y- components if the hydrodynamic forces acting on the hydrofoil be denoted by drag \( D \) and lift \( L \), then the complex forces acting on a hydrofoil calculated within the linearized theory are given by

\[
D+il = \int \{ P |_{y=0} - P |_{y=q} \} \ dt \tag{2.4}
\]

\[
L = - \rho U \int \gamma(x) dx \quad \text{D = \rho U \int \gamma(x) dx} \tag{2.5}
\]

The boundary condition on the hydrofoil it can be approximate to

\[
z(x) = \psi \left( \frac{U}{r} + \frac{1}{U} \right) v = \frac{1}{U} \left( \frac{\partial \Phi}{\partial y} \right) \tag{2.6}
\]

where \( u, v \) are the components of liquid velocity as follows:

\[
u = - \frac{\partial \Phi}{\partial x} - \frac{\partial \Phi}{\partial x}, \quad v = - \frac{\partial \Phi}{\partial y} - \frac{\partial \Phi}{\partial y} \tag{2.7}
\]

Using [2.2] we can write [2.6] in the form

\[
z(x) = \frac{1}{2\pi U} \int \frac{a}{(\xi-x)} + k(\xi-x)) \ dt \tag{2.8}
\]

with

\[
k(x-x') = \frac{(x-x')^2}{(x-x')^2 + 4h^2}, \quad 2v \int \frac{a}{(x-x')^2 + 4h^2} \sin \psi + 2h \cos \psi \ ds \tag{2.9}
\]

III. The Optimum Shape of Hydrofoil of Minimum Drag

We pose the problem of minimizing the drag coefficient subject to a constraint on curvature, \( K \), together with a constraint on the length of the hydrofoil as follows:

\[
L(\gamma, z, o, x) = 0, \quad \int a \int \Phi(x,y) = 0, \quad \int a \int k(x,y) dx \tag{3.1}
\]

subject to

where \( y, z \) are related by \( \gamma \) and \( \lambda, \lambda_2 \) are Lagrange multipliers.

We define an admissible function as any function \( \gamma(x) \) which satisfies the Hôlder condition (\( r<1 \)) and we define the optimal function as an admissible function which minimize

The Necessary Condition Of Optimality. \( \delta L(\gamma, z, o, x) = 0 \),

which yields:

\[
z(x) = \frac{1}{2\pi U} \int \frac{a}{(\xi-x)} + \frac{\lambda_1(z)}{\sqrt{1+\lambda_2^2}} \tag{3.11}
\]

This equation is a necessary condition for the existence of an external \( \gamma(x), z(x) \), combines with the integral equation, \( [2.8] \), to give a pair of singular integral equations which are to be solved for \( \gamma, z \) subject to appropriate conditions and constraints.

Substituting from [2.8] in [3.6], we obtain

\[
\frac{1}{2\pi U} \int \gamma(x) dx = \frac{\lambda_1(z)}{\sqrt{1+\lambda_2^2}} \tag{3.7}
\]

We consider the solution of [3.7] for the slope \( z(x) \) only in the case of small slope, and we approximate to [3.7] as follows:

Now, we use the method of iteration to solve equation [3.8] as follows:

We introduce function sequences of the form

\[
z_0, z_1, z_2, ..., \gamma(z_1, z_2, ...) \tag{3.9}
\]

and the stages in the iteration procedure would be as follows:

(a) First we solve

\[
\int \{ \int [2\lambda_2 z_1(s) \cdot \lambda_1 z_1(s)] ds, \quad a \int \]u \int [2\lambda_2 z_1(s) \cdot \lambda_1 z_1(s)] ds, \quad a \int \]

for \( \gamma_0, z_0 \) which gives

\[
z_0(x) = F_0 \sin \left( \frac{\gamma_0}{\sqrt{\lambda_2}} \right) \tag{3.11}
\]

where \( F_0, \gamma, \text{Co is an arbitrary constant and } m_0 = \frac{\gamma}{2\lambda_2} \)

(b) Secondly, we solve

\[
\int \{ \int [2\lambda_2 z_1(s) \cdot \lambda_1 z_1(s)] ds, \quad a \int \]u \int [2\lambda_2 z_1(s) \cdot \lambda_1 z_1(s)] ds, \quad a \int \]

where \( K_0 \), Co is an arbitrary constant and \( m_0 = \frac{\gamma}{2\lambda_2} \)
4. KOTCHIN, N.E. 1951. "On the wave-making resistance and lift of bodies submerged in water". The Society of Naval Architects and Marine Engineers. Trinity Place, New York 6, N.Y.
A DIRECT SIMULATION OF THE FLOW AROUND A CIRCULAR CYLINDER
SINUSOIDALLY OSCILLATING AT LOW KEULEGAN-CARPENTER NUMBERS

Papolu Manikyala Rao, Kunio Kuwahara and The Institute of Space and Astronautical Science,
Sagamihara-Shi, Kanagawa, 229, Japan,

ABSTRACT- A finite difference simulation method is presented for the viscous flow field around an arbitrarily moving boundary. Numerical solutions are obtained by directly integrating the incompressible Navier-Stokes equations of finite difference form by adopting a moving grid system, based on a time dependent coordinate transformation. Evolution with time of the flow structures induced by a circular cylinder performing sinusoidal oscillation in a fluid at rest, by means of vortex shedding, is studied at Keulegan-Carpenter number, $Kc = 9.4$. The time dependent drag and lift are also explained.

INTRODUCTION

The study of two dimensional oscillatory flows is of great importance in the design of cylindrical structures such as offshore platforms. Information about the fluctuating forces on an oscillatory cylinder is of special interest to fluid dynamicists and offshore engineers. Several experimental investigations have been carried out on fluid structure interaction problems, a good account of which are available in Bearman, et. al.(1985), Williamson(1985), Sarpkaya(1986) and Tatsuno and Bearman(1990).

BASIC EQUATIONS AND NUMERICAL METHODS

In the present study, we introduced the following generalized transformations of coordinates, which includes the time variables, in order to deal with moving boundary effectively,

$$\xi^i = \xi^i(x, y, t), i = 1, 2, 3$$

where $x, y, t$ are the variables in the physical domain and $(\xi^i, i = 1, 2, 3)$ are the variables in the computational domain. The transformation of coordinates and matrix are written as

$$z = z(\xi^1, \xi^2, \xi^3), T_{ij} = \frac{\partial x^i}{\partial \xi^j}, J = \det(T_{ij})$$

where $J$ is the Jacobian. The metric tensor $g^{ij}$ is given by

$$g^{ij} = T^{ik}T^{jk} \delta_{kl}, g = \det(g_{ij}), g^{ij} = (1/2)g^{-1}e^{imn}e^{lpq}g_{pq}g_{mn}$$

where $\delta$ and $e$ denote the Kronecker delta and the Eddington permutation symbol, respectively.

Consider a circular cylinder oscillates in a viscous incompressible fluid, in the direction parallel to $x$-axis. Instantaneous velocity of oscillation is given by

$$V = V_m \sin(2\pi t/T)$$

where $V_m$ is amplitude of oscillatory velocity, $T$ period of oscillation and $t$ time. The $\xi-S$ equations for the sinusoidal flow are written as

$$\frac{1}{Kc} \frac{\partial u^i}{\partial t} + (u^j u^i)_{,j} = -g^{ij} P_{,j} + \frac{1}{Kc} g^{ik} u^k - \frac{2\pi}{Kc} \cos(2\pi t) g^{ij} T^2$$

$$u^i_{,t} = 0$$

REFERENCES

EXTENSION OF THE $\lambda$-FORMULATION TO IMPERFECT GAS FLOWS

D. LENTINI

Dipartimento di Meccanica e Aeronautica
Università degli Studi di Roma "La Sapienza"
Via Eudossiana 18, I-00184 Roma RM, Italy

ABSTRACT

The $\lambda$-formulation for compressible flows is extended to gas flows with specific heat varying as a function of the temperature. The proposed formulation is based on a suitably specified function of the temperature, which allows defining the Riemann variables as linear combinations, with constant coefficients, of the dependent quantities. The resulting equations are solved via a fast solution algorithm. A test case is worked out of a nozzle flow, evidencing the differences with respect to the solution for constant specific heat.

I. INTRODUCTION

The $\lambda$-formulation (Moretti 1979, 1987) has proved to be a powerful tool for the numerical solution of compressible flows of perfect gas. For 'perfect' we mean that the gas is assumed to be both thermically and calorically perfect. The first attribute refers to its obeying the perfect gas law, while the latter denotes that its specific heat $c_p$ is taken as a constant.

This formulation has been successfully extended to finite-rate chemically reacting flows (Lentini and Onofri 1986, 1987 A and B), but still with the limitation that the component gases of the reacting mixture are perfect.

While the assumption of thermally perfect gas is closely approximated in virtually all cases of practical interest, the requirement that the gas (or the component gases) is calorically perfect may be unsatisfactory to describe flows subjected to large temperature excursions. Indeed, $c_p$ exhibits a fairly large variation with temperature. For example, the specific heat of air varies by about 25% in the temperature range $300-2000^\circ K$.

In this paper we extend the $\lambda$-formulation to (inert) flows of thermally perfect gas with $c_p$ varying as a function of the temperature $T$. Such gases are sometimes referred to as imperfect gases. The resulting formulation is extremely simple and involves a limited computational overhead over the perfect gas case.

It is applied here to the computation of the flow in a quasi one-dimensional nozzle in order to prove the workability of this approach and to evidentiate the differences with respect to a perfect gas computation. However, it will be apparent that the range of application of the present formulation is completely general.

II. FORMULATION

The formulation is presented here for simplicity for a quasi onedimensional flow.

We assume as state variables the speed of sound $a$ and the entropy $s$, and the velocity $u$ as the motion variable. Consequently, the continuity and momentum equations, which we write for convenience in the form

$$
a \frac{\partial u + u \partial s}{\rho} + a \partial s = -a u \frac{A_z}{A} \tag{1}
$$

$$
u_1 + u v_z + \frac{1}{\rho} p_z = 0 \tag{2}
$$

will be recast in terms of the variables $u, a, s$. To this end, we observe that the speed of sound is related to the temperature $T$ via the relationship

$$
a^2 = \gamma RT \tag{3}
$$

$\gamma$ being the gas specific heats ratio and $R$ its constant. Upon logarithmic differentiation we obtain

$$
2 \frac{a'}{a} = \left(1 + \frac{\gamma T}{\gamma} \right) \frac{T'}{T} \tag{4}
$$

Here the prime denotes differentiation with respect to either $t$ or $z$, $T'$ is the derivative of $\gamma$ with respect to $T$. From the first principle of thermodynamics, written for adiabatic flows with pressure work only

$$
e_{u} T' = RT p' \rho \tag{5}
$$

one gets, in view of (4)

$$
p' = \delta \left(1 + \frac{\gamma T}{\gamma} \right) \frac{a'}{a} \tag{6}
$$

with the position $\delta = (\gamma - 1)/2$. This expression will be substituted in the continuity equation.

The pressure gradient term in the momentum equation can be expressed by means of the thermodynamic relationship

$$
\frac{1}{\rho} p = h' - T s' \tag{7}
$$

where $h'$ is the enthalpy; then, after eq. (4), with $h' = c_p T'$

$$
\frac{1}{\rho} p = \delta \left(1 + \frac{\gamma T}{\gamma} \right) a a_z - T s_z \tag{8}
$$

The set of eqs. (1, 2) can then be recast as

$$
\frac{1}{\delta \left(1 + \frac{\gamma T}{\gamma} \right)} a_z + u a_z + a u_z = -a u \frac{A_z}{A} \tag{9}
$$

$$
u_1 + u v_z + \frac{1}{\delta \left(1 + \frac{\gamma T}{\gamma} \right)} a a_z - T s_z = 0 \tag{10}
$$

At this juncture, we make the crucial remarks that the terms $a'$ can be expressed as $a \gamma T'$, being $a$ a function of the temperature $T$, and that the term

$$
\frac{1}{\delta \left(1 + \frac{\gamma T}{\gamma} \right)} a_T = \frac{1}{\gamma - 1} \sqrt{\frac{RT}{T}} = \frac{c_p}{a} \tag{11}
$$

is solely a function of the temperature as well. We can thus define the function

$$
F'(T) = \frac{\gamma \frac{c_p}{T}}{\gamma - 1} d\theta \tag{12}
$$

where $\theta$ is the temperature as a running integration variable, and $T^0$ is an arbitrary reference temperature. Notice that the integrand is always positive so that the function $F'(T)$ is monotonic and can easily be inverted. This definition allows expressing the terms in the derivative of $a$ in eqs. (6,7) as

$$
\frac{1}{\delta \left(1 + \frac{\gamma T}{\gamma} \right)} a_T = F_i \tag{13}
$$
and similarly for the derivative with respect to \( x \). We can then recast

\[ F_1 + u F_x + a u_x = \beta \]  
(9)

\[ u_t + u u_x + a F_x - T s_x = 0 \]  
(10)

with \( \beta = -a u \lambda_2/A \). By summing and subtracting eqs. (9,10), and

\[ \lambda_1 = u + a \quad \lambda_2 = u - a \]

we get the final form

\[ R_{1t} + \lambda_1 R_{1x} - T s_x = \beta \]  
(11)

\[ R_{2t} + \lambda_2 R_{2x} + T s_x = \beta \]  
(12)

having defined the Riemann variables as

\[ R_1 = F + u \quad R_2 = F - u \]

Thus, the new variable \( F \) allows the definition of Riemann variables for imperfect gas flows as linear combinations, with constant coefficients, of the dependent quantities, in analogy with the formulation for perfect gas. In the case of isentropic, strictly one-dimensional flow, such variables represent true Riemann invariants.

Further, the imperfect gas formulation becomes formally identical to the perfect gas one. It is apparent that the present formulation can be extended without any difficulty to multidimensional flows, by redefining the Riemann variables in a similar fashion.

### III. SOLUTION ALGORITHM

As an example of the present formulation, here we apply it to the flow in a converging-diverging nozzle. We limit our analysis to steady state solution only. A semi-implicit algorithm, developed along the guidelines of Moretti's (1983) fast solver, is used. In this iterative technique eqs. (11,12) are integrated separately, at each step, by successive sweeps all over the computational domain.

The discretized form of eq. (11) is, with second-order upwind differencing:

\[ \frac{R_{1,n} - \hat{R}_{1,n}}{\Delta t} + \frac{\lambda_1 R_{1,n} - R_{1,n-1}}{\Delta x} = \bar{\beta} \]

being \( \lambda_1 \) always positive in the flow under consideration; the averages are defined as

\[ \bar{\lambda}_1 = \frac{\lambda_{1,n} + \lambda_{1,n-1}}{2} \quad \bar{\beta} = \frac{\beta_n + \beta_{n-1}}{2} \]

The caret denotes the previous iteration level.

As far as eq. (12) is concerned, we have to make a distinction between the cases \( \lambda_2 > 0 \) and \( \lambda_2 < 0 \). In the former case (supersonic flow) the discretization is analogous to that for \( R_1 \):

\[ \frac{R_{2,n} - \hat{R}_{2,n}}{\Delta t} + \frac{\lambda_2 R_{2,n} - R_{2,n-1}}{\Delta x} = \bar{\beta} \]

with

\[ \bar{\lambda}_2 = \frac{\lambda_{2,n} + \lambda_{2,n-1}}{2} \quad \bar{\beta} = \frac{\beta_n + \beta_{n-1}}{2} \]

whereas for \( \lambda_2 < 0 \) (subsonic flow) we get

\[ \frac{R_{2,n} - \hat{R}_{2,n}}{\Delta t} + \frac{\lambda_2 R_{2,n+1} - R_{2,n}}{\Delta x} = \bar{\beta} \]

Astride the sonic line the accuracy of the algorithm is reduced to first-order, by setting

\[ \bar{\lambda}_1 = \lambda_{1,n} \quad \bar{\lambda}_2 = \lambda_{2,n} \quad \bar{\beta} = \beta_n \]

in order not to violate the domains of dependence of the variables. The eq. in \( R_1 \) is integrated by sweeping in the positive \( x \)-direction, whereas the one in \( R_2 \) is integrated by sweeping from the sonic line to the exit (supersonic region) and from the sonic line to the inlet (subsonic region). This algorithm gives a very fast convergence to the steady solution.

For the case of imperfect gas, the procedure is initialized by first guessing the temperature field, and accordingly estimating \( F \) via the relationship

\[ F = F(T) \]  
(13)

as in (8). The \( a \) field is analogously initialized by means of the relationship

\[ a = a(T) \]  
(14)

as in (3). This allows computing \( \lambda_1 \) and \( \lambda_2 \). Then, at each iteration step the new values of \( R_1 \) and \( R_2 \) are computed, and \( u \) and \( F \) are updated as

\[ u = \frac{R_1 - R_2}{2} \quad F = \frac{R_1 + R_2}{2} \]

The speed of sound is then recomputed as

\[ a = a(F) \]  
(15)

where the argument of (14) has been transformed into \( F \) by inverting eq. (13). \( \lambda_1 \) and \( \lambda_2 \) can then be recomputed and a new iteration cycle started, until convergence is attained.

The boundary condition at the inlet involves matching the total enthalpy

\[ h_0 = h + \frac{u^2}{2} \]

where \( h \) is recovered as

\[ h = h(F) \]  
(16)

with the stagnation enthalpy \( h_s \), i.e.:

\[ h_s = h(R_2 + u) + \frac{u^2}{2} \]

where \( F \) is expressed via the value of \( R_2 \), computed from downstream. The above nonlinear relationship is iterated to get the value of \( u \) at the inlet.

Once the converged solution is obtained, the density can be recovered as a function of \( F \), via eq. (5):

\[ \log \frac{\rho}{\rho_0} = \frac{1}{R} \int_0^T \frac{c_s^2}{\rho} d\theta \]  
(17)

the pressure is then computed by means of the (thermal) equation of state.

In the present implementation the functions defined by eqs. (13-17) are computed off-line and approximated by 4th-order fits obtained by means of orthogonal polynomials, in order to make the computational procedure as straightforward as possible.
IV. RESULTS

We show here a comparison between a calculation performed for perfect gas (with \( \gamma = 1.4 \)) and the present approach for imperfect gas, for a critical flow in a converging-diverging nozzle. The working fluid is assumed to be air, with chosen properties as a function of the temperature accounted for by means of the so-called NASA polynomials (e.g., see Gardiner 1984).

A nozzle with conical converging (semi-angle 45°) and diverging (semi-angle 15°) sections, matched by a throat section with a conical profile, is considered. The inlet and throat radii are 0.10 and 0.12, respectively, the radius of curvature of the throat is equal to the throat radius, and the geometric expansion ratio (exit to throat area) is 5. The stagnation temperature \( T_0 \) is assumed to be 2000°K (note that for imperfect gases results do not scale with temperature). Although at this high temperature some molecular dissociation and formation of nitric oxides do occur, a chemical equilibrium computation (performed with the code by Reynolds 1981) shows that the cumulative mass fraction of the ensuing products is less than 1%, and accordingly, in this study we neglect effects related to the varying composition. In particular, the variation in the average molecular weight turns out to be absolutely negligible.

A computational grid with 40 nodes is chosen so, for perfect gas, it gives a relative discrepancy between the Mach number computed analytically and by the numerical solution at convergence limited to a maximum (all over the computational domain) of 0.7%.

Figure 1 compares axial profiles of velocity \( u \) for the two cases of perfect and imperfect gas. The nozzle profile is superimposed on the figure.

![Fig. 1. Axial profiles of velocity \( u \) (made dimensionless with the reference value \( \sqrt{RT_0} \)). Dash-dotted line, perfect gas; dashed line, imperfect gas; solid line, nozzle contour.](image1)

![Fig. 2. Axial profiles of Mach number \( M \). Dash-dotted line, perfect gas; dashed line, imperfect gas; solid line, nozzle contour.](image2)

![Fig. 3. Axial profiles of temperature \( T \) (made dimensionless with the reference value \( T_0 \)). Dash-dotted line, perfect gas; dashed line, imperfect gas; solid line, nozzle contour.](image3)

![Fig. 4. Decay of the residual of the computation. Dash-dotted line, perfect gas; dashed line, imperfect gas.](image4)
In the same fashion Fig. 2 compares axis of profiles of Mach number, and Fig. 3 profiles of temperature. It can be observed that while and M are affected to a limited extent by the varying c, the temperature computed with the imperfect gas model exhibits a much slower decay than the perfect gas solution, owing to the higher specific heat. In particular, the gap between the two solutions at the outlet is about 200 K, thus underlining the need to account correctly for the caloric properties of the gas.

Further, neglecting the effects of temperature on c, leads to overestimating the mass flow rate by 4.1%, and the velocity thrust by 1.5%; accordingly, the specific impulse is underestimated by 2.5%. Fig. 4 compares the convergence history of the computations for perfect and imperfect gas. The solution is with describing the same (arbitrary) Mach number distribution in both cases. It is apparent that the solution algorithm described in Sect. III gives an extremely fast convergence rate, with the steady state reached in as few as 30 iterations.

V. CONCLUSIONS

A simple formulation to effectively extend the formulation to imperfect gas flows is presented and tested.

REFERENCES


The construction method of the perfect-absorbing boundary conditions on tw. domains boundary, on which the fluid motion is described by linear, and in another one—by nonlinear shallow water equations, have been described. The approach proposed also permits to solve the problem on a free wave pass through the outer boundary out of the domain and the problem on the reflected waves outgoing from the domain through the boundary, on which the disturbances are preset as boundary conditions.

1. With numerical simulating of long waves propagation in water areas one often deals with the following, at the first sight "different" problems:

Problem 1. Let's describe the long waves propagation from the source to the shore. It's known, that non-collapsing wave motion along the deep ocean is rather well described by the linear shallow water equations, whereas at the wave outgoing to the shallows, where non-linearity effects become significant, the non-linear equations should be applied. While solving this problem it's expedient to divide all the flow region into two subregions and on the line, dividing them, to preset the boundary conditions, passing the wave from the "deepwater" region to the "coastal" one without reflection. It's clear, that the question is only about non-physical reflections, caused by different models "joint".

Problem 2. Let's describe the wave motion in the bay, caused by the fact, that on the boundary, faced to the sea, the wave train is preset as the boundary condition. Suppose, that the bay bottom is such, that the reflected wave, going to meet the bay entering one, begins to form rather early. Thus, here the reflected wave out of the region through the boundary, on which the disturbance is assigned.

Problem 3. With the application of finite-difference methods for the wave processes simulation in the open water reservoirs the calculation is performed in the finite region. Thus, here occurs the problem on the free pass of the waves, coming to the calculated region boundary. The boundaries, possessing the property to pass any disturbances without reflection, we should call "transparent" or "absorbing". These boundaries have no physical essence and their consideration is connected with the method of the numerical calculations performance.
These formulas are considered to be the basis for finite-difference algorithms construction while solving the problem about the solution joint on two mediums boundary. To construct the boundary conditions for the problems of the complete wave outgoing through the outer boundary the auxiliary technique has been applied. Its main point is the extension of the water filled domain by the channel with constant depth bottom, where the flow is described by linear shallow water equations. This procedure results in the following boundary conditions at the point $x=x_*$, where $x=x_*$ is the left domain boundary:

$$
\begin{align*}
 u|_{x=x_*} &= 1/2(r+s), \\
 \zeta|_{x=x_*} &= (r-4(gd_*)^{1/2})^2/(16g) - d_*, \\
 r &= (g/d_*)^{1/2} \zeta, \\
 d_* &= \left| \right|_{x=x_*}.
\end{align*}
$$

Where $r=r(X_*,t)$ is $r$-invariant, describing the wave, which will outgo out of domain bounds.

The main point in the corresponding numerical algorithms is $r$- and $s$-invariant approximate construction in boundaries neighborhood.

During the construction of analogous boundary conditions for the two-dimensional problem the assumption of the possibility of such approximate flow description, that in the limits of the small discrete time intervals the wave "front" should be straight and keep its motion direction, was made. For this case the application of one-dimensional algorithm worked out without difficulties. For the two-dimensional problems was a success.

The series test calculations, demonstrating the workability of the method proposed and a good accuracy of the numerical algorithms, has been performed. Fig.2,5 illustrate the solutions of the problem on the wave outgoing out of the domain. The waves surfaces for the sequential time moments have been shown.

Abstract—The impulsively started flow past a circular cylinder in a uniform freestream for Reynolds numbers, Re from 200 to 500 is computed by a time splitting method to get the lift, drag, and Strouhal number.

1. INTRODUCTION

The discrete vortex method is a long established approach for modelling the convection of vorticity in two-dimensional problems of separated flow past bluff bodies. See Gerrard[1] and Sarpkaya[2]. This method, which can now also diffuse into account, has recently been extensively reviewed by Sarpkaya[3]. In the time-splitting method, the processes of convection and diffusion are treated separately. Whereas Chorin[4] and Smith and Stansby[5] use random walk methods to model the diffusion, Benson et al.[6] introduced and Burrows[7] developed a vorticity re-distribution method in an attempt to reduce the number of discrete vortices required by the model and hence reduce the magnitude of the computational task. In [6], after extensively testing the parameters of the model at Re=40 (based on diameter), the method was applied to flow past circular cylinders for Re=20, 40, 100, and 200. The present work describes further development and validation of this approach for Re up to 500.

2. MATHEMATICAL AND COMPUTATIONAL FORMULATION.

The Navier-Stokes equations for an incompressible fluid are expressed in non-dimensional form in terms of the vorticity, \( \zeta \), and the stream function, \( \psi \), as follows

\[
\frac{\partial \zeta}{\partial t} + (\mathbf{u} \cdot \nabla) \zeta = \frac{1}{Re} \nabla^2 \zeta
\]

\[
\nabla^2 \psi = -\zeta
\]

(1)

where \( t \) is the time and \( \mathbf{u}=(u,v) \) are the velocity components in \( (x,y) \) Cartesian coordinates with origin at the centre of the cylinder and the \( x \)-axis in the freestream direction. Assuming \( \zeta \) is known at a typical instant, equation (2) is solved for \( \psi \) as follows. A radially expanding cylindrical polar mesh \((R(\cdot),\Theta(\cdot))\) about the cylinder is transformed into a uniform rectangular mesh \((\Theta(\cdot),R(DT))\) by the equation

\[
R(DT)=\exp(C1.RD(DT)/DRD)
\]

(3)

where

\[
RD(1)=0;
\]

\[
RD(DT)=RD(1+1)-RD(1) \quad \text{for} \quad 1,2,3,\ldots,\text{NR}
\]

(4)

and there are also NR mesh points in the \( i \)-direction of length \( DTH \), where

\[
DTH=2\pi/(\text{NR}-1)
\]

(5)

Then the transformed equation (2) is solved for \( \psi \) on the rectangular mesh using the Poisson solver of Le Bail[8], which requires that \( \Psi \) is of the form \( 2^m \cdot \). \( \text{NR} \cdot 129 \) is found to be a good compromise between resolution and economy. After choosing the radial extent of the flowfield. \( R(\text{NR})=80 \) say. (3) gives \( C1=0.30423 \). \( R(\text{NR}) \) must be sufficiently large to ensure that the grid always contains all the vorticity for the duration of the computation. \( DRD \) is a constant chosen to be the same order as \( DTH \). Next, the velocity field is computed. Then the known vorticity field is discretised onto point vortices which are convected by Eulerian integration for a small convection timestep. \( DTT \). This procedure is Reynolds number independent and so the mesh and other computational parameters should be invariant as \( Re \) changes.

While convecting, it is assumed that these point vortices are diffusing as Oseen vortices. After a time \( t \), a zero age Oseen vortex will diffuse a radial distance \( 4.836\sqrt{t/Re} \) (see Slouti[9]) and so as the vortex spreads, each part of it will move with an increasingly different velocity and may become distorted. To overcome this problem the diffused vorticity of each vortex is re-discretised onto new zero-age point vortices every timestep. \( DTR \). The validity of this approach was appraised in test cases described in [6] and is satisfactory if \( DTR \) is large enough to allow vorticity to diffuse at least two mesh lengths. For a square mesh of length \( GS \), this gives

\[
DTR > 0.0995.Re.GS^2
\]

(6)

where Slouti's result above has been used.

The vorticity in the flowfield is located in two main areas. Firstly, a wake region extends downstream of the cylinder. This is conveniently embraced within a grid of width \(-5c\sqrt{5} \cdot \) length \(-3c\) with square mesh of length \( GS \). Here, \( GS=0.2 \) if, for example, \( Re=200 \), then (6) implies a minimum diffusion timestep of 0.8 in this mesh.

Secondly, there is a boundary layer region of high vorticity around the cylinder of approximate thickness \( BL \), say at the shoulders of the cylinder where

\[
BL = 3.82/Re
\]

(7)

(See [7]). When, for example, \( Re=200 \), \( BL=0.27 \) and when \( Re=500 \), \( BL \) decreases to 0.17. Although this region is contained within the rectangular mesh, a finer body fitted mesh allowing more resolution of the diffusion near the surface and a diffusion timestep, \( DTR \), say, lower than for the rectangular mesh is required in order to calculate the aerodynamic forces on the cylinder. In [4], the polar mesh defined for the convection is also used to apply the redistribution algorithm near to the body. This grid has mesh size \( 0.034 \times 0.049 \) on the body expanding to \( 0.042 \times 0.062 \) at radius 1.27. Now in timestep \( DTT \), vortices in the coarser rectangular mesh may diffuse through the finer polar mesh to the surface of the body. To prevent this, all vortices within radius 1.4 are diffused using the fine polar mesh. Since for \( Re<200 \) there are at least 8 mesh points within \( BL \), the convection mesh satisfactorily duplicates the diffusion. For \( Re=500 \), however, there are only 5 mesh points within \( BL \) and so a new and separate polar mesh for diffusion near the body is now introduced. It has the same form as equations (3.4.5) but with \( DTH \) halved. For equal mesh lengths in the radial
and circumferential directions, C1 is reduced to Cl=0.024034. This gives more mesh points within BL. Then for Re<200 there are at least 8 mesh points within BL. Then DTI has a minimum value of Re/10^4. This is calculated from (6) based on the maximum mesh length at radius 1.4 in the polar grid.

Now the cost of the diffusion algorithm increases as the square of the number of mesh points a vortex diffuses in a timestep. For a fixed mesh, this limits the magnitude of the timestep. For the rectangular mesh, DTR is chosen so that a vortex diffuses through no more than 3 mesh lengths. Then DTR < 0.2239.Re.GS^2 (8)

Similarly, for the polar diffusion mesh, the timestep is limited so that vortices diffuse through no more than 3 mesh lengths at radius 1.4. Now the mesh length is smaller on the cylinder surface and so for the above restricted timestep, the algorithm must actually allow diffusion over 5 mesh lengths near the surface. Methods of satisfying the no-slip condition on the boundary, reflecting vorticity from the surface of the cylinder, calculating the lift and drag coefficients and other details are described in [63] and [77].

3. VALIDATION AND DISCUSSION OF THE RESULTS.

Table 1 shows 6 different sets of parameters for which the computer code was run for Re=200. Their choice was guided by the above considerations. NTH2 is the number of mesh points in the circumferential direction for the inner polar diffusion mesh.

<table>
<thead>
<tr>
<th>Version</th>
<th>DTT</th>
<th>DTI</th>
<th>DTR</th>
<th>NTH2</th>
</tr>
</thead>
<tbody>
<tr>
<td>R200V1</td>
<td>0.2</td>
<td>0.2</td>
<td>1.0</td>
<td>257</td>
</tr>
<tr>
<td>R200V2</td>
<td>0.075</td>
<td>0.075</td>
<td>0.975</td>
<td>257</td>
</tr>
<tr>
<td>R200V3</td>
<td>0.1</td>
<td>0.2</td>
<td>1.0</td>
<td>129</td>
</tr>
<tr>
<td>R200V4</td>
<td>0.075</td>
<td>0.075</td>
<td>1.95</td>
<td>257</td>
</tr>
<tr>
<td>R200V5</td>
<td>0.05</td>
<td>0.1</td>
<td>1.0</td>
<td>129</td>
</tr>
<tr>
<td>R200V6</td>
<td>0.05</td>
<td>0.15</td>
<td>1.95</td>
<td>129</td>
</tr>
<tr>
<td>R400</td>
<td>0.075</td>
<td>0.075</td>
<td>1.95</td>
<td>257</td>
</tr>
<tr>
<td>R500</td>
<td>0.075</td>
<td>0.075</td>
<td>2.475</td>
<td>257</td>
</tr>
</tbody>
</table>

The diffusion in the inner polar mesh takes place every one or two convection timesteps whereas the diffusion in the outer rectangular mesh is much less frequent due to the larger mesh length. For the range of Reynolds number investigated, periodic oscillating wakes are obtained. The Strouhal number, St, average drag coefficient, Cd(avg) and range (twice the amplitude) of the drag, Cd(ra) and lift Cl(ra) coefficients are computed. These are shown in Table 2, with results of Brazer et al. [1] for Re=200.

<table>
<thead>
<tr>
<th>Version</th>
<th>Cd(avg)</th>
<th>Cd(ra)</th>
<th>Cl(ra)</th>
<th>St</th>
</tr>
</thead>
<tbody>
<tr>
<td>R200V1</td>
<td>1.35</td>
<td>0.10</td>
<td>1.66</td>
<td>0.194</td>
</tr>
<tr>
<td>R200V2</td>
<td>1.37</td>
<td>0.13</td>
<td>1.58</td>
<td>0.190</td>
</tr>
<tr>
<td>R200V3</td>
<td>1.38</td>
<td>0.11</td>
<td>1.55</td>
<td>0.190</td>
</tr>
<tr>
<td>R200V4</td>
<td>1.37</td>
<td>0.13</td>
<td>1.58</td>
<td>0.191</td>
</tr>
<tr>
<td>R200V5</td>
<td>1.36</td>
<td>0.12</td>
<td>1.57</td>
<td>0.191</td>
</tr>
<tr>
<td>Braza</td>
<td>1.32</td>
<td>0.12</td>
<td>1.60</td>
<td>0.190</td>
</tr>
<tr>
<td>R250</td>
<td>1.36</td>
<td>0.21</td>
<td>1.81</td>
<td>0.198</td>
</tr>
<tr>
<td>R400</td>
<td>1.36</td>
<td>0.22</td>
<td>2.26</td>
<td>0.206</td>
</tr>
<tr>
<td>R500</td>
<td>1.36</td>
<td>0.32</td>
<td>2.60</td>
<td>0.205</td>
</tr>
</tbody>
</table>

In [63], for Re<200, good agreement was found with other published results, which the present investigation also confirmed but the results of [63] were not so good at Re=200. The parameters of R200V1 correspond to those used in [63] and the present work suggests that the convection timestep is too large. This results in too large an average Cd and too small values of the ranges. When DTT<0.1, however, variation of the parameters subject increases as the square radius 1.4 in the polar grid.

Now the mesh length is smaller on the cylinder surface and so for the above restricted timestep, the algorithm must actually allow diffusion over 5 mesh lengths near the surface. Methods of satisfying the no-slip condition on the boundary, reflecting vorticity from the surface of the cylinder, calculating the lift and drag coefficients and other details are described in [63] and [77].

REFERENCES
NUMERICAL MODELING SEPARATION OF FLOW OF VISCOUS FLUID IN THE PIPES

OSTAPENKO V.A.
Faculty of Mathematics and Mechanics
State University of Dnipropetrovsk
pr. Gagarina 72
Dnipropetrovsk-10, 320625, U S S R

Abstract-It is proposed method for discovering separation regions of viscous fluid in bent pipes. That method is based on decomposition of full system equations of hydrodynamics of viscous fluid and consists in three stages. On the first stage the equations of streamlines are obtained and time- and space-average length streamline for given pipeline is calculated. Stagnating zones and reverse motions are obtained as well. On the second stage we construct one-dimensional equivalent model of viscous fluid motion along streamlines which takes into consideration the narrowing of area of transversal section by stagnating zones and reverse motions. Solving the initial-boundary value problem for equations one-dimensional equivalent model we obtain the subdomains of the pipeline at which are valid the necessary conditions for separation of flow.

The aim of third stage is the control of sufficient conditions for separation of flow. For that purpose the system equations of hydrodynamics for boundary layer of bent pipes the transversal section of which is near to circle is derived. For that system we solve initial-boundary value problem only in those subdomains of the pipeline at which are valid the necessary condition for separation of flow and thus check sufficient conditions for separation of flow.

The application of described method permits to construct the pipelines with small energy losses because it is possible to prevent separation of flow. The decomposition of separation of flow problem decreases considerably the expenditures of computing time.

I. INTRODUCTION

Under some circumstances of the motion of fluid in the pipes can occur separation of flow and that phenomenon leads to sharp change of pressure, speed and temperature of fluid compared with their values without separation. Most essential cause which leads to separation of flow is the form of channel especially sharp change of its transversal section [1].

The separation of flow leads to essential increase of the resistance of fluid motion in the pipes. Therefore constructing pipelines with small energy losses it is necessary to create such form of channel that separation of flow should not occur.

To solve the problem of creating pipelines in which the motion of fluid occurs without separation of flow it is not necessary at full measure to elaborate mathematical model of separation of flow. In that case it is necessary only to answer the question: occurs or does not occur separation of flow in given pipeline under given parameters of fluid which flows in pipe. Last problem is essentially simpler and its solution can be obtained by methods of hydrodynamics. At present paper is suggested the method for solution such problem.

From the point of view hydrodynamics for arising separation of flow two factors are necessary: positive gradient of pressure and viscosity of fluid [1]. It is known that in three-dimensional space separation of flow arises in boundary layer [1,2]. Exactly, the separation of flow occurs when nearest to wall of channel the streamline comes off wall. Therefore the necessary condition for separation of flow is

$$\frac{\partial u}{\partial n} \mid _S = 0$$

where \(S\) is contour of pipeline, \(n\) - unit vector of external normal to surface \(S\), \(q\) - normal to \(S\) component of fluid velocity. The necessary condition (1) can lead to separation of flow if in the same region pressure \(p\) increases towards the flow, i.e. is valid condition

$$\frac{\partial u}{\partial n} \mid _S > 0$$

The sufficient condition for separation of flow is

$$\frac{\partial u}{\partial n} \mid _S < 0$$

So to solve separation of flow problem we propose a method of decomposition which consists in three stages. Main idea of the first two stages is to represent outside to boundary layer flow as one-dimensional one along streamlines and with the help of such conception to find the parameters of outside flow. For that purpose we construct one-dimensional equivalent model of the motion of viscous fluid in pipe [4].

II. FIRST STAGE

It is known that equations of motion of perfect fluid are one-dimensional towards streamlines. In case of viscous fluid the equations of motion are one-dimensional towards streamlines accurate to term which takes into consideration the rotation of velocity field [4]. The full system equations of hydrodynamics of viscous fluid is singularly perturbed one and if \(\lambda = \lambda_0\), where \(\lambda\) and \(\lambda_0\) are constants of Lamé, becomes full system equations of hydrodynamics of perfect fluid. It is known in the theory of singularly perturbed boundary value problems, the differens between solutions of perturbed and non-perturbed problems has order \(O(\lambda^0)\) or \(O(\lambda^0)\) everywhere with the expection of boundary layer which thickness has the same order.
On the first stage of decomposition we find equations of streamlines for three-dimensional motion of fluid in pipe. Above considerations permit to find those equations by means of solving initial-boundary value problem for perfect fluid. After the velocity field for perfect fluid is found we calculate that one for time of activity of pipeline, calculate length of every average streamline and calculate average length of streamlines in pipeline.

II. SECOND STAGE

On the second stage there we construct one-dimensional equivalent model of viscous fluid motion in the pipeline. The length of one-dimensional region of motion is equal to average length of streamlines. The main task for creating the one-dimensional model is to take into consideration the influence of the forces of friction on the motion of fluid in pipeline. That model is not ordinary consequence of three-dimensional motion of viscous fluid under assumption that all functions which are inserted into equations depend on one space variable but is more complete one. It takes into account, in particular, the change of the area of transversal section and perimeter of pipelines towards the flow (along x-axis) and the heat exchange with the walls of pipeline and is described by following equations

\[
\begin{align*}
\frac{\partial}{\partial x} (\rho V S) + \frac{\partial}{\partial y} (\rho V^2 S) + \frac{\partial}{\partial z} (\rho V S) + \frac{\partial}{\partial t} V Q = 0 \\
\frac{\partial}{\partial x} (\rho S) + \frac{\partial}{\partial y} (\rho V S) = 0 \\
\frac{\partial}{\partial x} [\rho (S c_i + 3 V_i V_j)] + \frac{\partial}{\partial y} [\rho S (c_i + 3 V_i V_j)] + \frac{\partial}{\partial z} [\rho S (c_i + 3 V_i V_j)] + \frac{\partial}{\partial t} V Q = 0 \\
\rho = \rho_0 \sqrt{x/a}.
\end{align*}
\]

(4)

It should be noted that in one-dimensional model there are taken into consideration additionally the results of solving of three-dimensional problem on the first stage. By means of analysis of velocity field in the pipeline, in particular, we discover stagnating zones and reverse motions and using those results make correction of geometrical values of area of transversal section and perimeter of the boundary contour of pipeline.

As the result of solving initial-boundary value problem for system (4) we obtain the values which characterise motion of viscous fluid in pipeline. In particular, with the help of function of pressure \( p(x,t) \) we can discover sections of pipeline in which condition (2) is valid, i.e. such sections where separation of flow may occur in principle.

IV. THIRD STAGE

To answer the question whether separation of flow really occurs we have to solve three-dimensional initial-boundary value problem for full system of hydrodynamic differential equations. However, taking into consideration that separation of flow occurs only in boundary layer, it is sufficient to consider that problem only there, moreover only in those parts of boundary layer where condition (2) is valid. In its turn the small thickness of boundary layer permits to simplify equations of motion in it.

Let us consider equations of hydrodynamics of viscous fluid in cylindrical channel with circular cross-section of radius \( R \). Let us introduce cylindrical polar coordinates \( r, \theta, z \) and transformation \( \xi = R \xi - r \). We introduce as well the transformation to undimensional variables (with asterisk) with the help of scaling

\[
\begin{align*}
x &= \xi \xi^* \quad y = \xi y^* \quad z = \xi z^* \\
v_r &= \xi v_r^* \quad v_\theta = \xi v_\theta^* \quad v_z = \xi v_z^* \\
p_r &= \xi p_r^* \quad \rho_\theta = \xi \rho_\theta^* \quad \rho_z = \xi \rho_z^*
\end{align*}
\]

(5)

and conditions of connection

\[
\begin{align*}
v_r &= \xi v_r^* \quad v_\theta &= \xi v_\theta^* \quad v_z = \xi v_z^* \\
p_r &= \xi p_r^* \quad \rho_\theta &= \xi \rho_\theta^* \quad \rho_z = \xi \rho_z^*
\end{align*}
\]

(6)

Executing in full system of hydrodynamic differential equations all of these transformations, evaluating every term and neglecting the terms which are sufficiently small compared with unity, we derive system equations for boundary layer (again in dimensional variables)

\[
\begin{align*}
\frac{\partial}{\partial t} (\rho V S) + \frac{\partial}{\partial x} (\rho V V_S) + \frac{\partial}{\partial y} (\rho V V_y S) + \frac{\partial}{\partial z} (\rho V S) + \frac{\partial}{\partial t} V Q = 0 \\
\frac{\partial}{\partial t} (\rho S) + \frac{\partial}{\partial x} (\rho V S) = 0 \\
\frac{\partial}{\partial x} [\rho (S c_i + 3 V_i V_j)] + \frac{\partial}{\partial y} [\rho S (c_i + 3 V_i V_j)] + \frac{\partial}{\partial z} [\rho S (c_i + 3 V_i V_j)] + \frac{\partial}{\partial t} V Q = 0 \\
\rho = \rho_0 \sqrt{\frac{x}{a}}.
\end{align*}
\]

(7)

Now for the solution of separation of flow problem we have to solve initial-boundary value problem for system (7) in boundary layer. The motion of fluid outside boundary layer with the high degree of accuracy can be described by parameters of one-dimensional equivalent flow. After solving such problem we can verify the satisfaction of conditions (1) and (3). If those conditions are valid it means that form of pipeline have to be changed and for new form we have to solve again the same problem.

This technique can be applied as well for bent pipelines with near to circle cross-section. To take into consideration of centrifugal forces the first equation (7) has to be written as

\[
\frac{\partial}{\partial t} (\rho V S^*) + \frac{\partial}{\partial x} (\rho V V_S^*) + \frac{\partial}{\partial y} (\rho V V_y S^*) + \frac{\partial}{\partial z} (\rho V S^*) + \frac{\partial}{\partial t} V Q^* = 0
\]

and to right-hand side of second equation (7) must be added the term

\[
\frac{\partial}{\partial t} (\rho S^*) + \frac{\partial}{\partial x} (\rho V S^*) = 0
\]

(8)

where

\[
R = \frac{\partial}{\partial \theta} (\xi \xi^* + \xi^2 \xi \xi^* + \xi^2 \xi \xi^* ) + \frac{\partial}{\partial \theta} (\xi \xi^* + \xi^2 \xi \xi^* + \xi^2 \xi \xi^* )
\]

(9)

The method which has been proposed permits to create packaged programs for analysis separation of flow in the pipes. Using that programs we can design pipelines with small losses of energy because of that phenomenon there do not occur.

REFERENCES

COMPUTATIONAL ANALYSIS OF THREE-DIMENSIONAL SHOCK-WAVE/TURBULENT BOUNDARY LAYER INTERACTION

A. KOURTA and H. HA MINH

Institut de Mécanique des Fluides de Toulouse
I.N.P.T.- U.R.A. C.N.R.S. 0005
Av. du Prof. Camille Soulé, 31400 Toulouse-Cedex FRANCE

and

Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique (C.E.R.F.A.C.S.)
42, avenue G. Coriolis, 31057 Toulouse-Cedex FRANCE

ABSTRACT:

The topic of this paper is the prediction of three-dimensional flows and the analysis of shock wave/turbulent boundary layer interaction. For these, a two-step method based on the MacCormack scheme has been used. The averaged Navier-Stokes equations with turbulence model (Baldwin-Lomax) were solved numerically in a general-coordinate system for three dimensional turbulent flows. The numerical computation was performed for a 3D channel flow. The configuration is the same as the one studied experimentally at ONERA [1], [2] in a wind tunnel. It consists of a bump mounted in the lower wall. The countoured portion of the bump is at an angle of 60° with the upstream flow. The flow is then highly three-dimensional. The features of this flow are predicted. A detailed description of a 3D interaction and a complete analysis of the physical phenomena and the comparison with the experimental results are done.

I. INTRODUCTION

Several Navier-Stokes codes have been developed in order to calculate two or three dimensional viscous compressible flows. The three dimensional code is needed when we have to analyze complex fluid flows. The study of three dimensional flow has become a subject of significant importance in the aerodynamic.

The purpose of this investigation is to develop a code based on a MacCormack method for solving the compressible Navier-Stokes equations to solve three dimensional turbulent flows.

The aim of this study is to test the capability of this code to simulate 3D shock wave/turbulent boundary layer interaction. This phenomenon is one of the interest gas dynamic problems. The effects of such interaction are the increase of the aerodynamic drag and the possibility of buffet onset. The natural consequence of this interaction is the separation. Hence, experimental investigations [1, 2, 3] were done to provide a set of data and to analyze this interaction.

In our case, the numerical computation was performed for a 3D transonic channel flow. The configuration is the same as one studied experimentally by BENAY et al [1] in a wind tunnel. The flow is highly three dimensional with the strong shock wave/boundary layer interaction.

II. GOVERNING EQUATIONS

If we consider the case of homogeneous flow excluding chemical reactions or very high temperature effects, the full Navier-Stokes equations constitute a good physical model. This model describes conservation of mass, momentum and total energy. With the mass-weighted averaged Navier-Stokes equations, a turbulence model is used to relate the turbulent flux terms to the mean flow parameters, through the use of an eddy viscosity coefficient. The turbulent viscosity is obtained by the Baldwin-Lomax model [4].

III. NUMERICAL METHOD

The numerical method is an explicit version of a MacCormack scheme [5, 6]. The predictor and corrector structure is used. For each time step, an explicit increment is evaluated by using the forward or backward approximations for the inviscid part and the central difference for the viscous terms. The scheme is a finite volume, cell center method.

In order to improve numerical efficiency, the second order accurate flux splitting has been included in the method. The flux splitting was motivated by one need for a better description of discontinuities and a more rigorous treatment of the boundary conditions. The flux splitting used here is close to the one developed by Steger and Warming [6, 7].

IV. RESULTS

The numerical method, described above, has been applied to the calculation of the transonic flow in a three dimensional channel. The forth boundaries are treated as a no-slip boundaries. Given the total pressure and total temperature, the inlet flow angle and the outlet downstream static pressure, calculations are performed. The same conditions used in the experimental study were implemented. The total pressure is 92 kPa, the total temperature is 300 K and the Reynolds number at the sonic state with the throat width as length scale is $1.13 \times 10^6$.

The features of this flow are predicted. Figures (1) and (2) show iso-Mach lines in the longitudinal vertical planes, at two different sections. These plots reveal the existence of the lambda shock comprising of a front oblique
shock and a quasi normal rear shock. The second foot of the lambda shock is not very intense as it is in a two dimensional configuration. The system of the shocks is generally associated with a strong interaction entailing separation. The length of the separation zone in this case is smaller than in a two dimensional configuration. The results are in good agreement with the experimental data. The main difference with a 2D shock-wave/boundary-layer interaction is that in a 3D situation, the flow can take a transverse or crosswise direction, and consequently when separation occurs we generally do not observe recirculating bubble in the streamwise direction.

Isobar lines on the bump are shown in figure (3). It reveals a nearly two dimensional incoming flow which is turned by a shock forming slightly downstream of the top of the bump. In this figure the shock foot is observed. At the shock foot, interaction with the boundary layer takes place entailing separation. The three dimensional character of the flow can also be observed downstream of the top of the bump.

Comparison to experimental data is given in figure (4). In this figure we plot the Mach number profile in one section before the shock. The agreement seems to be good in the external inviscid flow. In the near wall regions, the calculation overestimates the thickness of the boundary layers. This is due to the overestimation of the turbulence viscosity and perhaps to the coarse mesh used.

V. CONCLUSIONS

A three dimensional code based on a Mac Cormack scheme has been developed. It has been applied to the calculation of transonic turbulent channel flow. The features of this three dimensional flow were predicted and the shock-wave/boundary-layer interaction was analysed. Due to the calculation of turbulent viscosity, the thickness of the boundary layers is larger than the experimental one.

Acknowledgments

The authors would like to acknowledge helpful discussions with Prof. D. Vandromme. Thanks also to T. Pot from ONERA for providing technical reports.

References

The Theory and Computation of the Second-order Water Wave Forces upon a 2-D Floating Body

Tang Ling
Shanghai Institute of Applied Mathematics & Mechanics
Shanghai University of Technology, Yan Chang Road, Shanghai 200072, PRC

Abstract: In this paper, the expressions of the second order water wave forces upon a 2-D floating body are derived with the perturbation method, and the numerical computations are carried out with the boundary element method at the same time. Compared with the results of the experiments and of the calculations done by other researchers, the present results are more reasonable and accurate.

I. Introduction

These years, with the rapid development of the ocean engineering, much attention was paid to the nonlinear water wave theory. This is because that the water wave forces are the most important loading to many marine structures such as ships and drug platform as well as some offshore structures. It is known that a body will be subjected to linear and nonlinear forces. In regular waves, we call the latter second-order wave forces which include two parts, the steady drifting forces with frequency equal to zero and the biharmonic oscillating ones with twice of the incident wave frequency. Both are very important factors and need full consideration.

Early in 1960, Haruo achieved a general expression for the horizontal steady drifting force of a 2-D body, which is proportional to the square of the reflected wave amplitude of the linear solution. His work was generalized later by Newman and Faltinsen. It is reported that there is a good agreement between the results of experiments and of calculations with this method. But it cannot predict the biharmonic wave force which also covers a considerable proportion in the total second-order wave forces.

In 1980s, a more direct investigation was carried out by Kyozuka. With the boundary element method, he obtained the solution of the diffractional and radiational problems in regular waves. Recently, the problem of a fixed 2-D body in regular waves was studied by Liu & Miao, they introduced the Dirac function to the inhomogenous free surface condition and found a complete and consistent solution for the second-order diffraction potential.

In this paper, the problem of a free body floating in regular waves is investigated. It is an profound research which is an extension of the previous work. The problems are solved respectively, which include the second-order diffraction of incident waves and the coupled effects of the first order motions of the floating body on second-order wave forces.

II. The Governing Equation and Boundary Conditions

As depicted in Fig.1, the two Cartesian coordinate systems are related each other by the following expressions:

\[ X = S_1 + x \cos(S_3) - y \sin(S_3) \]
\[ Y = S_2 + x \sin(S_3) + y \cos(S_3) \]

where \( S_1, S_2, S_3 \) are, respectively, the excursions in \( X \) and \( Y \) directions and rotation of the body about the origin \( O \).

We assumed that the fluid is incompressible and inviscid, the flow irrotational; thus the motion of the fluid can be described by a velocity potential \( \Phi \), and the governing equation is the Laplace equation:

\[ \nabla^2 \Phi(x, y, t) = 0 \]

Since the floating body is impermeable, we have

\[ \Phi_{n} = \Phi_{n0} \text{on the surface of the body} \]

Through analysis, we obtain the general free surface condition

\[ H_2 + g \Phi + 2i k \Phi + 2 \Phi \Phi + H_2 \Phi + \Phi \Phi + 2 i k \Phi \Phi = 0 \]

At the bottom of infinity, we have

\[ \lim \Phi_y = 0 \]

Besides, a radiation condition should be imposed at infinity, i.e., the diffraction waves must be propagating outward from the body.

III. Perturbation Analysis

For infinite deep water, the potential of incident wave \( H_2 \) can be expressed as

\[ H_2 = \frac{g(1)}{x} \exp(i k x + i k x - i \omega t) \]

in which \( \omega, k, a \) are wave frequency, wave number and wave amplitude respectively.

Let \( e = k a \) be the small wave slope, by means of perturbation technique, the potential has the form

\[ \Phi_{n} = \Phi_{n0} + e \Phi_{n1} + e^2 \Phi_{n2} \]

Since the amplitude of \( S_i (i = 1, 2, 3) \) are also very small, so we have

\[ \sin(S_i) = S_i + O(e^2) \]
\[ \cos(S_i) = 1 - 0.5 S_i^2 + O(e^4) \]

Thus,

\[ X = S_1 + x \cos(S_3) - y \sin(S_3) \]
\[ Y = S_2 + x \sin(S_3) + y \cos(S_3) \]

Similarly,

\[ n_2 = n_2 + \Phi_{n2} \]
\[ n_3 = n_3 + \Phi_{n3} \]

Since \( \Phi_{n0} = \Phi_{n01} + \Phi_{n02} \), the equations of different orders are

\[ \Phi_{III} = 0, \text{in fluid domain} \]
\[ \Phi_{IV} = 0, \text{at } y = 0 \]
\[ \Phi_{n0} = \Phi_{n01} \text{on } Co(x, y) \]

\[ \lim H_2 \Phi_y = 0 \]

and

\[ \nabla^2 \Phi_{n0} = 0, \text{in fluid domain} \]
Jiy-1kji=0, at y=0
Jin=pi, on Co(x,y)
Lim Jiy = 0
Lim (Jix + i k Ji) = 0

The first order problem can be solved with the Frank method numerically.

\[ \nabla H = 0, \text{in fluid domain} \]
\[ Hly - 4k Hl = 0, \text{at } y=0 \]
\[ (H1i) \phi = \phi(x,y)-H2n-Hdn, \text{on Co}(x,y) \]
\[ \text{Lim } Hly = 0 \]
\[ \text{Lim } (H1x + i k Hl) = 0 \]

\[ \nabla H2 = 0, \text{in fluid domain}\]
\[ H2y - 4k H2 = P1(x) \]
\[ \text{Lim } H2y = 0, \text{as } y \text{ approaches infinity} \]
\[ \text{Outgoing waves at infinity} \]

\[ \nabla H3 = 0, \text{in fluid domain}\]
\[ H3y - 4k H3 = P3(x) \]
\[ \text{Lim } H3y = 0, \text{as } y \text{ approaches infinity} \]
\[ \text{Outgoing waves at infinity} \]

where \( P = \left( iv/2g \right) \left( 2(kx^3 + hy^3) - \frac{iv}{3} (kx^3 - hy^3) \right) \), and
\[ P1 = -4k x^n \cos \left( \left( x + B/2 \right) \phi \right) \int \left( x + B/2 \right)^n dx \]
\[ P2 = -4k x^n \cos \left( \left( x - B/2 \right) \phi \right) \int \left( x - B/2 \right)^n dx \]

It is clear that \( H \) will be found out with the same process used in 1st order once the potential \( H2 \) and \( H3 \) has been solved. Here are the solutions of \( H2 \) and \( H3 \).

\( H2 = \frac{-sgn(x+B/2)}{(P1/4k) \exp(4ky + i4k) + P1/4k + sgn(x+B/2) \exp(-4ky - i4k)} \)
\[ \int \left( \frac{\cos^{(m+1)}(my)}{4k} \exp(-4kx + i4k) \right) dy \]

where \( x + B/2 = 1,0, \text{ when } x - B/2 \)
\[ sgn(x+B/2) = 0, \text{ when } x - B/2 \]

\( H2 \) approaches \( -P1/8, 3.1416 \) when \( x \) approaches 0.

\( H3 = \text{Re} \left( -i \left( 3.1416 \right) ^2 F2(s) \cdot ds \int \exp(-ivx + jvs) \right) \)
\[ \left( \text{outgoing wave} - \frac{\text{inflow}}{s} \right) \]

where \( z = x+jy \) and \( PV \) denotes Cauchy principle integrates.

IV. Equation of 1st order motions

According to theorem of Newton, we have
\[ N \left( \tilde{S}_1 - y \tilde{S}_3 \right) = \varphi F \]
\[ N \tilde{S}_2 = \varphi F + p\varphi - M \]
\[ N \tilde{S}_3 = \varphi F + p\varphi + Hg \]

where \( \varphi \) and \( y \varphi \) are mass and coordinate of the center of gravity in \( x+y \) axis and \( \int \varphi f(x+y) \varphi \) of the body.

In still water, the body is in equilibrium, that means \( g \varphi = p \varphi \). where \( p \) is the density of fluid. By using pressure equation we finally obtained
\[ \begin{bmatrix} -\varphi \left( H0d1 \right) + yC11, \varphi = 0, & -\varphi \left( D13-i \varphi \right) - yC13 \end{bmatrix} \]
\[ 0, \varphi \left( H0d2 \right) + p\varphi - iC22,0 \]
\[ -\varphi \left( D13-yC13 \right) + yC13, \varphi = 0, \varphi \left( D33-i \varphi \right) - p\varphi - p\varphi \]
\[ \left( Q1, Q2, Q3 \right) = \left( T1, T2, T3 \right) \]

where \( D_{ij} = \varphi \left( i/j \right) G_{ij} = \varphi \left( i/j \right) C_{ij} \), \( i=1,2,3 \)
\[ \varphi \left( T1 \right) = \varphi \left( T1 \right) + \varphi \left( T2 \right) \varphi \left( T3 \right) \]
\[ \left( T1, T2, T3 \right) \]

\[ AY = C.X \varphi \text{ ydy, } A \text{ is the submergder area of the body and } \varphi \text{ is the moment of inertial of wetted section about the } x \text{ axis.} \]

Since \( F = \text{Re} \left( e^{iv} \exp(-ivr) + e^{iv} \exp(ivr) \right) \) and \( F = -\varphi iC \left( H + yC \right) \) the expressions of forces with different order will be derived easily and is not listed here, readers may be referred to [7].

V. Results and Conclusion

As depicted in Fig.2, \( B=2.0(m), h=0.8(m), \lambda=0.76 \)
\[ Y_{fw} = 0.031, \text{Re}=0.667, I=108.30, k=1.0(1/m), \lambda=77.52(kg/m) \]
\[ \rho=102.8, g=9.81, Y_{fw}=0.829. \]

The computation results are showed in Fig.3 and Fig.4. It was carried out on the vax-11/750 digital computer in Shanghai.

From these computation, we may clearly found out that the present results seem agree rather well with the experimental ones. Thus, we may conclude that

1. The magnitude of biharmonic forces have the same order compared with the steady drifting forces, further more, they reveal the oscillating nature of high order forces.
2. The present theory can be used to predict the wave forces in irregular waves combined with Fourier analysis and application may also be found in 3-D calculation by means of method of stripping in seakeeping theory.

Reference

ON PERMANENT CAPILLAR-HEAVY WAVES IN FINITE CHANNELS

NABIL MOUSSA
Associate Professor and Mathematics Unit Head
The American University in Cairo, P.O. Box 2511, Cairo, EGYPT

Abstract—This work is based on the constructive existence proof of solutions of a comprehensive class of nonlinear free boundary value problems of plane hydrodynamics by E. Zeidler [5] and a general computational method for constructing the solutions numerically given by the author [3]. The case of permanent capillary-heavy waves in finite channels will be considered.

INTRODUCTION AND NOTATIONS

After formulating the boundary value problem in the stream plane in section 1, we apply Levi-Civita conformal mapping to get a nonlinear boundary value problem on the circular ring and transform the problem to an operator equation in the Banach space C_0 in section 2. In section 3 we formulate the statement of existence and uniqueness. The general computational method for constructing the solutions will be used in section 4 and in section 5 the solutions in different orders will be computed.

For the function \( f(\rho, \sigma) \) defined in the open circular ring \( E_0 = \{ z = \rho e^{i \sigma} : 0 < \rho < 1 \} \), the symmetry behaviour will be denoted by

\[ f(\rho, \sigma) = f(\rho, \sigma) \quad \text{for } f(1,\sigma) = f(\sigma). \]

For the Fourier series of \( f(\rho, \sigma) \)

\[ f = a_0 + \sum_{k=1}^{\infty} \sum_{\ell=0}^{\infty} \left( a_{k\ell} \cos \ell \sigma + b_{k\ell} \sin \ell \sigma \right) \]

we have the closure condition

\[ a_0(f) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f \, d\sigma = 0 \]

Further we define the following operators (see [1]):

\[ T_f = f - a_0(f) \quad \text{and} \quad J_f = T_f \, T_f \, d\sigma = 0 \]

\[ K_f = \frac{1}{2\pi} \int_{-\pi}^{\pi} f \, \cot \frac{\sigma - \sigma'}{2} \left( f(\sigma') - f(\sigma) \right) d\alpha' \]

\[ V_{qf} = \frac{2}{\pi} \sum_{k=1}^{\infty} \frac{q^k}{1 - q^{2k}} \int_{-\pi}^{\pi} \sin(k\sigma - k\sigma') f(\sigma') d\sigma' \]

\[ W_{qf} = -\frac{2}{\pi} \sum_{k=1}^{\infty} \frac{q^{2k}}{1 - q^{2k}} \int_{-\pi}^{\pi} \cos(k\sigma - k\sigma') f(\sigma') d\sigma' \]

and the following Stokes's parameters

\[ s_k = q^{-k} + q^k \quad \text{and} \quad d_k = q^{-k} - q^k \]

\[ t_k(q) = \frac{s_k}{d_k} = \frac{1 + q^{2k}}{1 - q^{2k}} \quad \text{and} \quad \eta_k(q) = k \, t_k(q) \]

\( Q_n(\sigma, \tau) \) will denote regular power series with power greater or equal to \( r \) in \( \tau \).

1. Problem Formulation

To describe permanent capillary-heavy waves of an ideal homogeneous incompressible liquid in a finite channel, we will follow the method of Levi-Civita [2] by considering the two systems of coordinates:

(i) The moving \( x-y \) system in which the surface seems to rest;

(ii) The \( X-Y \) rest system in which the liquid is resting on the ground (see Figure 1).

We have \( x=X+ct, y=Y \) with \( c \) the speed of propagation of the wave in the sense of Levi-Civita. The free surface \( y_0(x) \) satisfies

\[ \begin{align*}
(1) & \quad y_0(-x) = y_0(x) \quad y_0(x+\lambda) = y_0(x) \\
(2) & \quad \int_0^\lambda y_0(x) \, dx = 0 \quad -\lambda/2
\end{align*} \]

The flow potential \( W(z) = \psi + i\varphi \) in \( x-y \) system will be normalized through

\[ \psi(0) = 0 \]

As the free surface and the channel ground are flow lines, then

\[ \psi(x,y_0(x)) = 0 \quad \varphi(x,h) = -\varphi_o < 0 \]

with \( \varphi_0 > 0 \) the flow in the moving system.

For the velocity \( \vec{W} = u+iv \) let us assume, as Levi-Civita proposed, that

\[ \begin{align*}
(5) & \quad W(z) = c \, e^{-iW} \quad w = \theta + ix, \quad a_0(\tau, \sigma) = 0 \\
(6) & \quad \theta = \arg(u+iv), \quad c = \psi_o + i\varphi
\end{align*} \]

Consequently, we have the following boundary value problem:

Find \( y_0(x) \) satisfying (1) and (2) such that in the flow region \( G_0(z) = -h < y < y_0(x) \) there exists a holomorphic function \( W(z) = \psi + i\varphi \) satisfying (3) and (4). This function satisfies also

\[ \begin{align*}
(7) & \quad \psi = -h: \text{Im} W'(z) = 0, \quad i.e., \psi = 0 \quad \text{on the ground.}
\end{align*} \]

Along the free surface the Bernoulli's equation

\[ y = y_0(x) = gy_0(x) + 1/2 \left| W'(z) \right|^2 + p_i = K_o \]

is satisfied, where \( m \) is the density, \( \beta \) is the surface tension constant, \( p_i \) and \( p_o \) are internal and external pressures respectively.

2. Conformal Mapping and Back Transformation

Due to periodicity of \( W(z) \) we consider the strip

\[ G_{z'}: -\frac{\lambda}{2} \leq x \leq \frac{\lambda}{2}, \quad -h \leq y \leq y_0(x) \]

It follows that

\[ G_{z'}: -\frac{\lambda}{2} \leq \phi \leq \frac{\lambda}{2} \quad \psi_0 \leq \varphi \leq 0 \]

is the conformal mapping using \( W(z) \).

Then we map the strip \( G_{z'} \) using

\[ \zeta = \epsilon \sigma = e^{2\pi i W/c\lambda} \]

to the circular ring \( q = e^{2\pi i \psi_0/c\lambda} \leq |z| \leq 1 \) with a cut on the negative real axis (see Figure 1).

The back transformation will yield the free surface

\[ \begin{align*}
(9) & \quad x = \frac{1}{2\pi} \frac{c}{2} \int_{-\lambda/2}^{\lambda/2} e^{-i\sigma} \cos(\sigma) d\sigma \\
& \quad y = \frac{1}{2\pi} \frac{c}{2} \int_{-\lambda/2}^{\lambda/2} e^{-i\sigma} \sin(\sigma) d\sigma + y_0(0)
\end{align*} \]

where \( y_0(0) \) is to be determined using (2).
The Bernoulli's equation (8) will be transformed in 
$\zeta$-plane to

$$(10) \quad gn(\nu_0(0)-\frac{1}{2\pi r} e^{-r} \sin(e(0)) \sin(\theta)) + \frac{1}{2} e^{2r} e^{-r} \sin(e(0)) \pi_p = K_0$$

which is equivalent to

$$(11) \quad s_1(\lvert z \rvert = 1) : \text{and} \quad \frac{\partial}{\partial \theta} \frac{\partial}{\partial r} = \frac{1}{2}(e^{r} - e^{-r}) p e^{-r} \sin(e(0)) \pi_p = 0$$

with

$$(12) \quad s_q(\lvert z \rvert = q) : \text{and} \quad \frac{\partial}{\partial \theta} \frac{\partial}{\partial r} = \frac{2}{2\pi} \nu c^2 \sin(e(0)) \pi_p = 0$$

As $eU$, $rC$, we apply the operator $T$ to the integrand and get

$$(13) \quad s_i : \text{and} \quad \frac{\partial}{\partial \theta} \frac{\partial}{\partial r} = \frac{2}{2\pi} \nu c^2 \sin(e(0)) \pi_p = 0$$

Replacing the constant $a'$ in (12) by the functional $a'(r, \theta)$ and integrating using the operators $T$ and $J$, we get the operator equation

$$(14) \quad e = pJX_q + bJ^2 e + JM(b, \theta) = \lambda X_q$$

where $M(b, \theta)$ is an expression of the form $O_2(e, \theta)$. The linearized problem

$$(15) \quad e = pJX_q + bJ^2 e$$

has the solution

$$(16) \quad p = p_{n,b,q} = \frac{\pi^2 + b}{n^2(q)}$$

so $e = \sin n_0$, $n_1, 2, \ldots$

3. Statement of Existence and Uniqueness

Given $n, q, b$ with $0 < q < 1$ and $b \neq \frac{\pi^2}{n^2 - k^2}$ there exist numbers $r_0, s_0, t_0 > 0$ such that for given $s$ with $|s| < s_0$ exactly one solution $e \in C_r(U)$ of (14) exists which satisfies the condition

$$|e| \leq r_0, \quad b(n) = s, \quad \|p_{n,b,q} \| < s_0$$

For $s = 0$, $p$ is uniquely determined: $p = p_{n,b,q} = \frac{\pi^2}{n^2(q)}$. For $s = 0$ only the trivial solution $e = 0$ exists. The solution $e$ can be given as an absolutely convergent series in $C_r(U)$

$$(17) \quad e = \sin n_0 + \sum_{k=2}^{\infty} s^k \frac{n_k}{2^k}$$

As $s = -X_q n_0$, the series

$$t_k = \frac{n_k}{2^k}$$

converges absolutely in $C_r(U)$. Also

$$p = p_{n,b,q} + \sum_{k=2}^{\infty} s^k t_k$$

is absolutely convergent with $2m+1 = C$ (for the proof see [5]).

4. Construction of the Solutions

Using the method of Zeidler [5], we apply the operators $T$ and $J$ to the functional $a'(r, \theta)$ to get

$$a' = -1 + a = -1 + \sum_{k=2}^{\infty} s^k \frac{n_k}{2^k}$$

Our problem can then be formulated as:

$$e = pJX_q + bJ^2 e + JM(b, \theta) = \lambda X_q$$

Let $p_{n,b,q}$ be the pseudo-resolvent. Then we have

$$(18) \quad \tilde{n}_{n,b,q} = (1 - P_{n,b,q} JX_q - bJ^2 - P_n)^{-1}$$

with

$$P_{n,b,q} = b(n)(0)\sin n_0$$

It follows that

$$e = \tilde{n}_{n,b,q}$$

(20) with $b(n) = b(n)(0)\sin n_0$

5. Computation of the Solutions

We consider the case $n = 1$ of primitive waves, from which we can calculate the non-primitive waves $n = 1$.

From (22) we have

$$k(b, \theta) = \frac{k(b_{k+1} + b_{k-1})}{\pi_b}$$

(23)

and

$$b_{1}(b, \theta) = \lambda_{2}(b, \theta)$$

(24)

where $b_0(b, \theta) = 0$, $\pi_b = 2, 3, \ldots$

The algorithm consists of two steps:

(i) If $a_0(\lambda_2(b, \theta)) = 0$ and $a_0(\theta) = 0$, it follows from (23) that

$$a_0(f_{b}) = 0$$

(27)

from which we compute $a_b$.

(ii) If $b_0(b, \theta) = s$, we have

$$b_1(b, \theta) = 0, \quad \pi_b = 2, 3, \ldots$$

From which we compute $a_{s,1}$. Expanding $f(b, \theta)$ to the third order, gives

$$f = e^{e(\theta)} - \frac{1}{2} e^{e(\theta)} + \frac{1}{2} e^{e(\theta)} - \frac{1}{2} e^{e(\theta)} - \frac{1}{2} e^{e(\theta)} \cos 2 \theta - \frac{1}{2} e^{e(\theta)} \cos 2 \theta$$

(29)

with $p = p_0 + e$ and $e = (d_1/s_1)(1+b) + e$.

Now, we consider the first three orders of the solutions:

(a) First Order. From (29) we have $f_1 = p_0$. From (27) we have $a_{s,1} = 0$ and we already know that the linearized problem has the solution

$$a_1 = \sin 1, \quad t_1 = -1 \cos \theta$$

(30)

(b) Second Order. As $\epsilon = 0$ ($\epsilon_{2m+1} = 0$), we have

$$f_2 = -\frac{1}{2} a_2 + b_2 \cos \theta$$

From (27) we get $a_2 = -b_2 p_0$, consequently

$$f_2 = -(3b_1/4) \cos 2 \theta$$

Using (23) we get
\[
\theta_2 = \delta_2 \sin 2\alpha, \quad \tau_2 = -\gamma_2 \delta_2 \cos 2\alpha
\]

(31) with
\[
\delta_2 = \frac{3(s_3 + 3s_1)b}{2(d_3(b-2) - 3d_1(b+2))}, \quad \gamma_2 = \frac{s_2}{d_2}
\]

(c) Third Order. From (29) we have
\[
f_3 = -\varepsilon_2 \tau_1 \gamma \alpha_3 z \frac{3}{2} \alpha_2 \tau_1 \gamma - b \varepsilon_1 \tau_2 - \varepsilon_2 \theta_2 - \varepsilon_2 \theta_2 - b \varepsilon_1 \theta_1 \gamma = -\frac{p_0}{y_1} + b \varepsilon_2 \theta_2 - \frac{b}{y_1} \theta_1 \gamma + \frac{b}{y_1} \theta_1 \gamma - \frac{b}{y_1} \theta_1 \gamma
\]

After rewriting all products in terms of trigonometric polynomials, it is sufficient to consider the cos terms:
\[
f_3 = \cos \sigma (e_2 \gamma_1 - \frac{3}{4} \gamma_1 \delta_2 b + \frac{b}{y_1})
\]

Beside those terms there will be constants and terms in cos 3\sigma.

From (28) we get \( a_1(f_3) = 0 \) which yields
\[
\varepsilon_2 = \frac{3}{4} \delta_2 b - \frac{b}{y_1} \gamma_1
\]

The solution of our problem will then be
\[
\theta = s \sin \alpha + \delta_2 \sin 2\alpha + o(s^3)
\]

(33) \[
\tau = -\gamma_2 s \cos \alpha - \gamma_2 \delta_2 \cos 2\alpha + o(s^3)
\]

\[
p = \frac{1}{\tau_1} + \frac{b}{y_1} \gamma_1 - \frac{b}{y_1} \gamma_1 - \frac{b}{y_1} \gamma_1
\]

This result agrees with our result in the case of infinite channels when \( q = 0 \) (see [4]).

Through back transformation we can get all physical parameters as well as the equation of the free surface \( y_0(x) \).

Simple integration will yield
\[
y_0(x) = \frac{\lambda}{2\pi} (s \cos \frac{2\pi}{\lambda} x + \frac{1}{2} (s_2 - \frac{1}{2}) \delta_2 \cos \frac{4\pi}{\lambda} x + o(s^3))
\]

REFERENCES


COMPUTATION OF TRANSONIC FLOW BY PASSIVE BOUNDARY-LAYER CONTROL ON AN AEROFOIL.

by

R.K. Cooper, S. Raghunathan and J.L. Gray,
Department of Aeronautical Engineering,
The Queen's University of Belfast,
Belfast, Northern Ireland.

Abstract - A computational method used to calculate
the transonic flow over an aerofoil with passive
boundary-layer control is described briefly. Both
the computational and experimental results show
that passive control can reduce both viscous and
wave drag of aerofoils in transonic flow.

1. INTRODUCTION

Transonic flow over an aerofoil contains super-
sonic regions embedded in a subsonic field. The
supersonic flow invariably terminates in a shock
wave. The formation of shock waves on an aerofoil
gives an additional pressure drag (wave drag) to
the subsonic pressure drag. The shock wave also
imposes an adverse pressure gradient on the
boundary-layer.

A control technique which appears to show some
promising results for a substantial drag reduction
in transonic flow is the passive control of shock-
boundary-layer interaction (PCSB). This paper
presents a brief review of computational approaches
to the solution of transonic flow with passive
boundary-layer control.

2. THE CONCEPT OF PASSIVE CONTROL OF SHOCK-BOUNDARY LAYER INTERACTION

The concept of PCSB as originally suggested by
Bushnell and Whitcomb (see Ref.1) consists of a
porous surface and a cavity or plenum underneath
located in the region of shock boundary-layer inter-
action (Fig. 1). It is suggested that the static
pressure rise across the shock wave will result in
a flow through the cavity from downstream to
upstream of the shock wave. This is equivalent to
a combination of suction downstream and blowing
upstream of the shock. The cavity would also
increase the communication of signals across the
shock wave. These effects would lead to a rapid
thickening of the boundary-layer approaching the
shock which in turn should produce a system of weaker
shocks with an extended interaction region. This
will reduce the wave drag. The suction downstream
of the shock can also reduce separation and therefore
reduce viscous losses. The porous surface and the
cavity can also damp pressure fluctuations associated
with shock wave boundary-layer interaction.

The conservation form of the two dimensional
compressible Navier-Stokes equation in cartesian
co-ordinates without buoy forces or external heat
transfer can be written as

\[ \frac{\partial u}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = 0 \]  

where the vector equation comprises conservation
of mass, momentum and energy.

The approximations made to the above equations
to reduce computational grids and time in the order
of reducing accuracy are (i) thin layer approxima-
tions (TLA), (ii) parabolised Navier-Stokes
(PNS), (iii) Euler equations, (iv) full potential
flow equations and (v) small perturbation equations.
Whereas (i) and (i=) resolve viscous terms in the
flow to some degree of approximation, (iii) to (v) neglect the viscous terms. The viscous terms can
be resolved by coupling (iii), (iv) or (v) with equations based on boundary-layer approximations.

3. COMPUTATION OF TRANSONIC FLOW WITH PASSIVE CONTROL

Some of the methods mentioned above have been
used to compute transonic flow over an aerofoil
with passive control. These are described briefly
here.

3.1 Transonic small disturbance approach

The equation describing two dimensional invicid
irrotational transonic flow around a thin aerofoil
in transformed \( \xi, \eta \) co-ordinates in non conservative
form can be expressed as

\[ K_\xi^2 \xi^* + \frac{\phi^*}{\xi^*} - \phi^* \eta^* = 0 \]  

where \( \xi, \eta \) are transformed co-ordinates,
\( \eta^* = y/(ct) \)

The boundary condition on a solid surface is

\[ \phi^* = 0 \]  

Fig. 1. The concept of passive boundary-layer control
The equation can be solved by a successive line over-relaxation procedure with a central difference scheme in the subsonic region (elliptic) and upwind difference scheme in the supersonic region (hyperbolic). Savu and Trifu used this procedure to calculate the flow field around a porous aerofoil assuming that the normal velocities in the porous region obey Darcy's law

\[ v_\eta = \alpha \eta \sigma = \tilde{\sigma} \rho (\rho_0 \nu_\infty) \]

where \( \tilde{\sigma} \) is the porosity factor.

3.2 Full potential flow approach

Subsequently Chen et al. used full potential flow equations with the same boundary conditions used by Savu and Trifu to solve this problem. The results of the computation agreed qualitatively with those of Savu and Trifu. Further, it was shown that passive control can not only reduce drag but augment lift on an aerofoil.

3.3 Interactive boundary-layer approach

Refinement to the above solutions was achieved by introducing viscous effects. These include for the flow around an aerofoil with passive control, thin layer approximations by Chung et al., coupling full potential outer flow to a boundary-layer equation near the surface by Chung et al. and coupling transonic small perturbation theory to boundary-layer equations near the surface by Gray.

The viscous flow near the surface of the aerofoil and wake can be expressed by the following non-dimensional non-conservative steady two-dimensional boundary-layer equations in transformed \( \xi, \eta \) co-ordinates.

\[
(\rho u)_{\xi} \xi_x + (\rho v)_{\eta} \eta_x = 0 \tag{4}
\]

\[
\rho \left( (u_{\xi} \xi_x + u_{\eta} \eta_x) + v u_{\eta} \eta_y \right) = -\beta \rho u_{\xi} \xi_x + \left( \mu \frac{u_{\eta}}{\eta} \right) \eta_y \tag{5}
\]

\[
\rho c_p \left( u (T_{\xi} \xi_x + T_{\eta} \eta_x) + v T_{\eta} \eta_y \right) = \beta \rho \nu u_{\xi} \xi_x + (k T_{\eta} \eta_y) \eta + \mu (u_{\eta} \eta_y)^2 \tag{6}
\]

\[ p = p \eta, \quad \beta = \mu \rho u^2, \quad c_p = \gamma R/(\gamma - 1) \]

The boundary-layer equations were solved using MacCormack's predictor-corrector algorithm with appropriate initial conditions.

Baldwin-Lomax turbulence model was used for the transonic small perturbation boundary-layer interaction code.

Typical results obtained from these computations are shown in Fig. 2. The computational results agree qualitatively with the experimental results and show some of the features of transonic shock wave boundary-layer interaction with passive control. The porous surface splits the flow into a shock with the leading edge of the shock system anchored to the beginning of the porous region. This has the effect of reducing entropy changes across the shock and therefore the wave drag. It has also been shown that passive control can alleviate shock oscillations in the transonic flow.

![Fig. 2. Pressure distribution \( \tau = 12\% \) circular arc](image)

References


Abstract—We describe an efficient preconditioning technique for solving time dependent problems with local refinement both in time and in space. The preconditioner makes use of solvers on regular grids only, hence it can be easily incorporated in existing codes for solving parabolic problems based on standard time stepping. The difficulty that arises in local time stepping is that such a discretization always leads to nonsymmetric matrices. However, it turns out that we can precondition these problems with generalized conjugate gradient (GCG) type methods in an optimal way.

I. INTRODUCTION

We describe an efficient subdomain technique based on Domain Decomposition (DD) ideas for solving time dependent diffusion problems. We consider the following model problem

$$\frac{\partial u}{\partial t} - \nabla \cdot \left( \frac{k}{\mu} \nabla u \right) = f \quad \text{in} \quad \Omega \subset \mathbb{R}^2, \quad t \geq 0,$$

with appropriate boundary and initial conditions. The discretization of the problem can be done in the framework of the discontinuous Galerkin method, (cf. Thomée [6]) where we have a global time step \( \tau_g \). In the time interval \((t_n, t_{n+1}]\), \( t_n = n \tau_g, \quad n \geq 0 \), in regions \( \Omega \subset \Omega \) of special interest, we introduce a finer time step \( \tau_f = \tau_g / m \) for some \( m > 1 \). We also couple this with local refinement in space on the subdomain \( \Omega_2 \). See Figure 1 for a one dimensional domain \( \Omega \). Then we have intermediate time levels \( t_{n,m} = t_n + m \tau_f, \quad i = 0,1, \ldots, m \).

![Figure 1. Grid with local refinement in space and in time. At the interface \( \Gamma \times (t_n, t_{n+1}] \) 'slave' nodes (denoted by 'x') are introduced.](image)

By using a variational formulation (for details see Ewing et al. [5]) or by cell-centered finite difference approximation of the above problem, cf. Ewing et al. [3], one can derive a composite grid problem for the unknowns between two global time steps \( t_n \) and \( t_{n+1} \). We obtain a linear algebraic problem of the form, \( A \mathbf{x} = \mathbf{b} \). Note that, the composite-grid matrix obtained in this way is always nonsymmetric, but has a coercive symmetric part.

II. TWO-GRID PRECONDITIONER

Using the following DD ordering of the nodes in the time slab \( (t_n, t_{n+1}] \times \Omega \),

$$x = \begin{bmatrix} x_f \\ x_e \end{bmatrix} \Omega_2 \times (t_n, t_{n+1}]$$

we obtain the following 2 x 2 block form of \( A \),

$$A = \begin{bmatrix} A_f & B \\ C & D \end{bmatrix},$$

where the block \( A_f \) corresponds to a standard time stepping procedure on the subdomain \( \Omega_2 \times (t_n, t_{n+1}] \). That is, to solve systems defined by \( A_f \), we can use any available time stepping code on a rectangular grid.

The preconditioner, originally proposed in Bramble et al. [2] for elliptic problems with local refinement, and extended for time dependent problems in Ewing et al. [3], is constructed for the reduced problem

$$S \mathbf{x} = \mathbf{b} - CA_f \mathbf{b},$$

where \( S = D - CA_f^{-1}B \) is the reduced Schur matrix. Consider the problem on the original coarse-grid (i.e., without any local refinement in either space or time). Then we have a coarse-grid matrix \( \mathbf{A} \) defined for the coarse grid at the time level \( t = t_{n+1} \). Formally, using the same DD idea, we can partition \( \mathbf{A} \) in 2 x 2 block form as follows

$$\mathbf{A} = \begin{bmatrix} A_2 & B \\ C & D \end{bmatrix} \Omega_2 \times (t_{n+1}) \Omega_2 \times (t_{n+1}),$$

where \( A_2 \) is a block-matrix corresponding to a global time stepping but for the subdomain \( \Omega_2 \) only. Note that in this case \( A_2 \) is symmetric and positive definite, hence its reduced Schur matrix \( \mathbf{S} = \mathbf{D} - \mathbf{C} \mathbf{A}_2^{-1} \mathbf{B} \) is also symmetric and positive definite. The preconditioner for the reduced problem we choose is \( \mathbf{S} \). Note that in order to solve a system with \( \mathbf{S} \), we can use solvers for the matrix \( \mathbf{A} \), hence we can use any available software for time stepping procedures on rectangular grids.

In Ewing et al. [4] it was shown that using an approximate \( \mathbf{S} \) in a GCG-type method (since \( \mathbf{S} \) is nonsymmetric) will give an optimal convergent method. The convergence properties are independent of possible jumps of the coefficient \( k/\mu \) and the coarse-grid sizes. For more details and some numerical experiments we refer to Ewing et al. [1,5]. For application in industrial reservoir simulation codes, see Boyett et al. [1].

REFERENCES


A DOMAIN DECOMPOSITION PROCEDURE FOR PARABOLIC EQUATIONS

CLINT DAWSON
Department of Mathematical Sciences
Rice University
Houston, TX 77251-1892 U.S.A.

Abstract: A domain decomposition procedure for solving parabolic partial differential equations is presented. Error estimates are stated, and numerical results on a parallel processing machine are discussed.

I. INTRODUCTION

Domain decomposition procedures have received much attention in recent years as a way of numerically solving partial differential equations on parallel processing machines, see, for example, [1]. Much of this work has been directed at elliptic equations. These techniques are often extendable to implicit time discretizations of parabolic equations, since an "elliptic" equation must be solved at each time step, generally resulting in an algorithm requiring iteration between subdomain problems and interface problems. In the work described here, we avoid this complication by using information from the previous time step to calculate fluxes along an interface between subdomains. These approximate fluxes are then used as boundary data for subdomains. These approximate fluxes are then used as boundary data for

Let \( \Omega \) denote a spatial domain in \( \mathbb{R}^d \). Denote by \( \mathcal{L}^p(\Omega) \) and \( \mathcal{W}_0^p(\Omega) \) the standard Sobolev spaces on \( \Omega \), with norms \( ||\cdot||_p \) and \( ||\cdot||_{0,p} \), respectively. Let \( \mathcal{L}^p(\mathbb{R}) \), \( p = 2, \infty \), denote the standard Banach spaces, with \( ||\cdot||_p \) denoting the \( \mathcal{L}^p \) norm.

Let \( [\alpha, \beta] \subset [0,T] \) denote a time interval, \( X = X(\Omega) \) a normed space. To incorporate time dependence, we use the notation \( || \cdot || \left| \left| \cdot \right| \right|_{\Omega(t)} \) to denote the norm of \( X \)-valued functions \( f \) with the map \( t \mapsto f(t) \) belonging to \( \mathcal{L}^p([\alpha, \beta]) \).

First, consider the case \( d = 2 \) and \( \Omega = (0, 1)^2 \). Decompose \( \Omega \) into two subdomains; \( \Omega_1 = (0, \frac{1}{2})^2 \times (0, 1) \) and \( \Omega_2 = (\frac{1}{2}, 1)^2 \times (0, 1) \). Let \( \Gamma \) denote the interface between these two domains, \( \Gamma = \{ \frac{1}{2} \}^2 \times (0, 1) \), and let \( u \) satisfy

\[
\begin{align*}
(u_t - \Delta u + u &= 0, & \text{on } \Omega \times (0, T), \\
\frac{\partial u}{\partial n} &= 0, & \text{on } \partial \Omega \times (0, T),
\end{align*}
\]

where \( n \) is the outward normal to \( \partial \Omega \).

Our domain decomposition procedure relies on calculating an approximate normal derivative of \( u \) on \( \Gamma \). Let \( \phi \) be \( \phi_\Gamma(x) = \phi_\Gamma(x - (1/2)H) \), where \( H \) is a parameter, \( H > 0 \), and

\[
\phi_\Gamma(x) = \begin{cases} 1 - \epsilon, & 0 \leq x \leq 1, \\
\epsilon + 1, & -1 \leq x \leq 0, \\
0, & \text{otherwise.}
\end{cases}
\]

For \( \phi \) a smooth function, define an approximate normal derivative of \( \psi \) at each point \( y \) on \( \Gamma \) by

\[
\psi_\Gamma(x; y) = B(\psi_\Gamma(x; y)) = -\int_0^1 \phi(x; y) \psi(x, y) \, dx,
\]

and note that

\[
||\psi_\Gamma(x; y) - B(\psi_\Gamma(x; y))|| \leq CH||\psi_\Gamma||_{0}. \quad (6)
\]

Let \( \delta = \delta_0 < \delta_1 < \cdots < \delta_M = T \) be a given sequence; \( \Delta t^n = t^n - t^{n-1} \). We approximate \( u \) at time \( t^n \) by \( U^n \), where \( U^n |_{\Omega_j} \in \mathcal{M}_j \), and \( \mathcal{M}_j \) is a finite dimensional subspace of \( \mathcal{L}^2(\Omega_j) \). Let \( \mathcal{M} \) be the subspace of \( \mathcal{L}^2(\Omega) \) such that \( u |_{\Omega_j} \in \mathcal{M}_j \), and let \( \delta \) denote the jump in function values of \( u \) across \( \Gamma \) (which is well-defined). The approximations \( U^1, \ldots, U^M \) are given by

\[
\frac{2}{\delta t^n} \sum_{j=1}^{2} \left( \phi_\Gamma(U^n; \delta_j) + D_j(U^n, v) \right) + (B(U^n))_\Gamma = 0, \quad v \in \mathcal{M}. \quad (7)
\]

Here \((\cdot; \cdot)\) denotes the \( L^2(\Gamma) \) inner product,

\[
D_j(U^n, v) = (\nabla U^n, \nabla v)_\Omega + (U^n, v)_\Omega,
\]

and \( \partial_i U^n = (U^n - U^{n-1})/\Delta t^n \). Note that once \( B(U^{n-1}) \) has been calculated, \( U^n \) can be computed on \( \Omega_1 \) and \( \Omega_2 \) completely independently.

In order to state an error estimate for the above algorithm, let \( W \in \mathcal{M} \) be the elliptic projection of \( u \), defined for each \( t \in [0, T] \) by

\[
\frac{2}{\delta t^n} \sum_{j=1}^{2} \left( \phi_\Gamma(W - u; \delta_j) + D_j(W - u, v) \right) = 0, \quad v \in \mathcal{M}. \quad (8)
\]

Let \( \eta = u - W \). The following theorem is proved in [2].

Theorem 1. Suppose that the solution \( u \) is sufficiently smooth and that \( u^n \in \mathcal{M} \) is taken to be \( W \), \( u^n \), \( \Delta t \). Then there is an constant \( C \) independent of the spaces \( \mathcal{M}_j \), such that

\[
\max_n ||u^n - U^n|| \leq C \left( \frac{\delta t^n}{H} + \Delta t \int_0^T ||\eta(t)||_H \left( t \in (0, T) \right) \right),
\]

provided that \( \Delta t \leq H^2 \). The algorithm can be extended to domains \( \Omega \subset \mathbb{R}^d \) with processes uniformly smooth, Lipschitz boundaries, where the interface \( \Gamma \) is a uniformly smooth, (\( d - 1 \))-dimensional manifold. Moreover, different functions other than \( \phi \) may be used in the flux approximation \( B \), which give higher order accuracy in \( H \). In particular, suppose \( \phi(x) = \phi_\Gamma(x - (1/2)H/H) \), where

\[
B(x; \psi) = \begin{cases} \frac{(x - 2/12)}{2}, & 1 \leq x \leq 2, \\
\frac{x}{12} + 7/16, & 0 \leq x \leq 1, \\
\frac{x}{12} + 1/6, & -1 \leq x \leq 0, \\
\frac{x}{2} + 1/12, & -2 \leq x \leq -1, \\
0, & \text{otherwise.}
\end{cases}
\]

In this case, for \( \psi \) five times differentiable in \( x \),

\[
||\psi_\Gamma(x; y) - B(\psi_\Gamma(x; y))|| \leq CH. \quad (10)
\]

The proof of Theorem 1 relies on the following coercivity property of the method. Namely, for \( \psi \in \mathcal{M}, \) let

\[
\frac{1}{2} \sum_{j=1}^{2} D_j(\psi, \psi) + \frac{1}{H} \| \psi \|_H^2,
\]

and

\[
||\psi||_H^2 \leq C \left( \sum_{j=1}^{2} D_j(\psi, \psi) + (B(\psi, \psi))_\Gamma \right). \quad (12)
\]

In general, assume that for some \( H > 0 \), \( B \) is a linear map of \( L^2(\Omega) \) into \( L^2(\Gamma) \) and that it satisfies the following four conditions.

(i) There is a constant \( C_1 \) such that

\[
||B(\psi)||_H^2 \leq C_1 H^{-2} ||\psi||_{0}. \quad (13)
\]

(ii) There is a constant \( C_2 \) such that

\[
||B(\psi)||_H^2 \leq C_2 H^{-1} ||\psi||_0. \quad (14)
\]

(iii) There is a constant \( C_3 \) such that

\[
||B(\psi)||_H^2 \leq C_3 H^{-1} ||\psi||_0. \quad (15)
\]

(iv) There is a constant \( k \geq u \) and a constant \( C_3 \) which depends on the solution \( u \) such that

\[
||\frac{\partial u_\Gamma}{\partial n} - B(\psi, \psi)_{\Gamma}||_n \leq C_3 H^k, \quad (16)
\]

for \( 0 \leq t \leq T \), where \( \partial u/\partial n \) is the normal derivative of \( u \) on \( \Gamma \), in the direction from \( \Omega_1 \) to \( \Omega_2 \).

671
We note that for \( \phi = \phi_2, \Omega \) the unit square on \( \mathbb{R}^2 \), and \( \Gamma = \{0\} \times (0, 1) \), \( C_0 = 1.7, C_1 = 2, \) and \( C_2 = 1 \). For \( \phi = \psi, C_0 = 1.64, C_1 = 3.14, \) and \( C_2 = 8/3 \).

In general, assuming conditions (i)-(v) hold, we have the following Theorem [2].

**Theorem 2** Suppose that \( u \) is sufficiently smooth and that \( u^0 = W(, 0) \).

Let \( \Delta t = \max_n \Delta t^n \). Then there exists a constant \( C \), independent of the number of subdomains, such that:

\[
\max_n \| (u^n - u^m) \| \leq C \left( \Delta t + H^4 \right) \| \eta(t, t) \|_{\infty},
\]

where \( \eta \) is the residual below \( H^4 \) (i.e., the solution is computed up to \( H^4 \)).

\[
\Delta t \leq \frac{H^2}{\xi_0 C_0},
\]

**III. NUMERICAL EXAMPLES**

We have implemented the algorithm outlined above on a 12 processor Intel iPSC/860 at the Center for Research in Parallel Computing, Rice University. We now present some timings which demonstrate the parallelism of the method.

The first problem tested was:

\[
u_t + \Delta u = 0, \quad \text{on } \Omega \times (0, T),
\]

\[
\left. u(\cdot, 0) = \cos(x \pi) \cos(n \pi), \right\} \quad \text{on } \Omega,
\]

\[
\frac{\partial u}{\partial n} = 0, \quad \text{on } \partial \Omega \times (0, T),
\]

with \( \Omega \) the unit square on \( \mathbb{R}^2 \). This equation has solution \( u(x, y, t) = e^{-\frac{x^2}{2} t^2} u(x, y) \).

In our domain decomposition procedure, once the interface fluxes are calculated, a system of linear algebraic equations must be solved within each subdomain. Preconditioned conjugate gradient with diagonal preconditioning was used to solve these subdomain problems.

In Table 1, we present timings for runs with different types of domain decompositions. In column 1 of Table 1, the notation "m x n" refers to a decomposition of \( \Omega \) into \( m \) subdomains in the \( x \) direction, and \( n \) in the \( y \) direction. The decomposition was done so that the subdomains contained the same number of unknowns. In these runs, \( \Delta t = .0025 \) and \( H = .10 \). The final time \( T = .10 \). The approximating space in each subdomain was the tensor product of continuous piecewise linear functions in \( x \) and \( y \). An underlying rectangular, uniform, 80-by-80 grid was used. We present both the CPU time for each case and the average number of conjugate gradient iterations. The latter number was computed by summing the number of conjugate gradient iterations (per subdomain) required to "solve" the linear algebraic system at each time step, and dividing this quantity by the number of time steps. ("Solving" the linear algebraic system meant reducing the residual below \( 10^{-4} \).)

We observe that as the number of processors was doubled, the run time essentially decreased by a factor of two when the number of subdomains remained the same.

The second problem we considered was:

\[
u_t + \Delta u = 0, \quad \text{on } \Omega \times (0, T),
\]

\[
\left. u(\cdot, 0) = 0, \right\} \quad \text{on } \Omega,
\]

with

\[
\frac{\partial u}{\partial n} = \begin{cases} 1, & \text{on } (0, 1) \times (0, T), \\ -1, & \text{on } (1/4, 1) \times (0, T), \\ 0, & \text{otherwise}. \end{cases}
\]

This problem corresponds to having a heat source on the left boundary of \( \Omega \) and a heat sink on the bottom boundary. The results in this case are given in Table 2. For this problem, the average number of conjugate gradient iterations varied from one subdomain to the other. In Table 2, we present the maximum of these averages. This is the quantity which controls the problem, since data cannot be passed from one subdomain to another until the linear algebraic problems in each subdomain have converged. As in the previous case, the computation time is reduced by a factor of two with each doubling of processors. A factor of greater than two is obtained when the number of subdomains is doubled.

**REFERENCES**


Very Sparse Preconditioned Conjugate-Gradient on Massively Parallel Architectures

Serge G. Petiton

Laboratoire MASI
Université P. et M. Curie (Paris 6)
75252 Paris Cedex, FRANCE

Christine J. Weill-Duflos

Institut de Calculs Mathématiques
Université Paris 7
75251 Paris Cedex, FRANCE

and

Department of Computer Science
Yale University
P.O. Box 2158, Yale Station
New Haven, CT 06520, USA

ABSTRACT

Computing Sparse Linear Algebra problems on SIMD massively parallel architectures is an important and major challenge for the future of these machines. In order to obtain good performance when solving many large sparse linear algebra problems, we use a communication compiler which permits to obtain a good speed-up compared to classical global communications. We study in this paper a polynomial preconditioned conjugate gradient method to solve a symmetric positive definite very sparse linear system on massively parallel architectures. We evaluate performance obtained on CM-2 with respect to a few very sparse matrices. We conclude that to solve some large very sparse matrix algebra problems we can now rely on CM-2 performance as fast as could be obtain on vector machines for dense problems a few years ago.

1 Introduction

Researches in massively parallel algorithms are numerous. Especially since the introduction of the Connection Machine 2 (CM-2) by Thinking Machines Corporation. Sparse computations are not yet very developed despite the supercomputer target-applications developed on such problems.

The sizes of the concerned sparse matrices are sometimes very large and the degree of row (resp. column), i.e. the number of non zero elements by row (resp. column), is often small. Many scientific research fields generate now very large linear algebra problems with row degrees smaller than one per thousand of the matrix order. Thus, in scientific computing we often need to solve a linear system starting with very large sparse matrices. We shall study in this paper unstructured very sparse matrices, i.e. with very small degree of row and column.

Iterative methods are often used to solve these problems on SIMD massively parallel machines. These methods often use 3 major operations: sequential scalar operations, reductions (principally inner products) and matrix vector operations (principally matrix vector multiplication). Scalar operations are often made redundant on each processor to optimize communications between processors.

Very Sparse Preconditioned Conjugate-Gradient on Massively Parallel Architectures

Very Sparse Preconditioned Conjugate-Gradient on Massively Parallel Architectures

Very Sparse Preconditioned Conjugate-Gradient on Massively Parallel Architectures

Very Sparse Preconditioned Conjugate-Gradient on Massively Parallel Architectures

Very Sparse Preconditioned Conjugate-Gradient on Massively Parallel Architectures

Very Sparse Preconditioned Conjugate-Gradient on Massively Parallel Architectures

Very Sparse Preconditioned Conjugate-Gradient on Massively Parallel Architectures

Very Sparse Preconditioned Conjugate-Gradient on Massively Parallel Architectures

Very Sparse Preconditioned Conjugate-Gradient on Massively Parallel Architectures

Very Sparse Preconditioned Conjugate-Gradient on Massively Parallel Architectures

Very Sparse Preconditioned Conjugate-Gradient on Massively Parallel Architectures

Very Sparse Preconditioned Conjugate-Gradient on Massively Parallel Architectures

Very Sparse Preconditioned Conjugate-Gradient on Massively Parallel Architectures

Very Sparse Preconditioned Conjugate-Gradient on Massively Parallel Architectures

Very Sparse Preconditioned Conjugate-Gradient on Massively Parallel Architectures

Very Sparse Preconditioned Conjugate-Gradient on Massively Parallel Architectures

Very Sparse Preconditioned Conjugate-Gradient on Massively Parallel Architectures

Very Sparse Preconditioned Conjugate-Gradient on Massively Parallel Architectures

Very Sparse Preconditioned Conjugate-Gradient on Massively Parallel Architectures

Very Sparse Preconditioned Conjugate-Gradient on Massively Parallel Architectures

Very Sparse Preconditioned Conjugate-Gradient on Massively Parallel Architectures

Very Sparse Preconditioned Conjugate-Gradient on Massively Parallel Architectures

Very Sparse Preconditioned Conjugate-Gradient on Massively Parallel Architectures

2 Sparse Matrix Computation on the CM-2

On many iterative linear algebra methods when the matrices are sparse, the matrix-vector multiplication is the only computation that manipulates sparse objects [4]. Thus, we first focus on the massively parallel sparse matrix-vector multiplications.

Methods to compute the multiplication of a sparse matrix by a vector on massively parallel architectures principally depend on the matrix and vector mapping and on the number of processors. Let us assume that n is the matrix order and C the degree of row that we suppose equal to the degree of column. Thus, with Cn processors we can map the sparse matrices with scan class as described by P. Kumar [8], with only one element of the matrix by virtual processor, see [7] for detailed implementation on the CM-2.

The idea is to compute all the floating point multiplications of the matrix-vector operation in massively parallel mode and to reduce with addition the other operations in a log2(C) theoretical complexity. But we need two different mappings to do that efficiently. Then, we need to change the mapping during the matrix-vector operation.

The multiplication will be done with a mapping in which the element a(i,j) will be mapped on the virtual processor (ic,jc), ic = 0, C – 1 and jc = 0, n – 1, where ic is the row indice on the Compress Sparse Column (CSC) format representation of the matrix, see [3] for the description of the sparse matrix formats. Then, if we want to reduce with addition, along only one dimension, the partial results in a log arithme complexity, we need to re-map the partial results, stored on virtual processors (ic,jc) to the virtual processor (jc,ic), where jc is the column indice of the element a(i,j) on the Compress Sparse Row (CSR) format representation. Thus, the complexity of this column oriented sparse matrix vector multiplication is 1 + log2(C) operations plus one global one-to-one general send operation.

With the general CM-2 router, the time to do these irregular communications is really a strong bottleneck, except for very well adapted sparse matrix patterns. Fast sparse computation is really impossible on the CM-2 without using special tools to optimize these one-to-one general communications. Dennis Dahl developed such a tool [2] which permits us to obtain more than a speed up of 10 compared to the CM-2 general router [7]. Then, we can really do sparse computations on the CM-2.

It is an important and difficult goal to propose test sparse ma-
trices to evaluate the performance for massively parallel algorithm computations. In this paper we do not propose special application patterns. We try to find bounds on the performance. These ones depend principally on the global send and especially on the distance between the diagonal of the sparse matrices and the \( C \) non-zero elements. Then, we take matrices with \( C \) non-zero diagonals around the main diagonal (\( C \)-diagonal matrix pattern) and we perturb at random them with a given probability \( q \), which is a parameter of the evaluation. Thus, it is nothing but the exchange of non-zero elements \((i,j)\) of the \( C \)-diagonal with the elements \((i,j_{\text{rand}})\), where \( j_{\text{rand}} \) is a random integer number between \( 0 \) and \( n-1 \). Each of these perturbations of the \( C \) band matrix is done with the probability \( q \). Hence, we can study the performance with respect to \( q \). We also use matrices with non-zero elements uniformly distributed all along the rows and columns, with a non-zero main diagonal (\( C \) distributed matrix pattern). In the first case we will obtain close-to-peak performance when \( q = 0 \), i.e. without perturbation of the \( C \) diagonals, for this algorithm and in the other case we will have the lower performance.

But, this tool uses the CM-2 as a hypercube of the vector accelerators. As we have just one vector accelerator per each 32 physical processors set, for example on a 16K CM2 we have 512 of these floating point fast units, can we conclude that it is always massively parallel computation?

### 3 Algorithm and Implementation

We are interested in the conjugate gradient method for the resolution of the linear system \( Ax = b \) where \( A \) is a very sparse symmetric positive matrix. To improve the conjugate gradient method, it is possible to combine it with a polynomial preconditioning, see [1] and [6].

The polynomial \( s \) we chose for the preconditioning, is given by the Neuman serie:

\[
s(A) = I + B + \cdots + B^m \quad \text{where} \quad B = I - A \quad \text{and} \quad ||B|| < 1.
\]

The through-put, in term of Megaflops (Mflops), does not depend on the coefficients of the polynomial but on the degree.

We map the vector and scalar on a 2 D \( C \times n \) grid compatible with the mapping of the matrix and to optimize communications. We often compute with redundancy; for example the inner product is computed on each row of processors and distributed on each processor. And, we do not compute the polynomial directly but we use the matrix-vector multiplication to compute \( s(A)x \) at each iteration of the algorithm.

Likewise the matrix-vector multiplication, we take the same test matrices to obtain the performance of the algorithm. Figure 1 shows the performance on a 16K CM-2 when the degree of the polynomial is 4 and the matrix coefficients are real. We observe that the performance decreases as \( q \) increases and that the uniformly distributed pattern matrix is really a lower-bound. The performance looks like the matrix-vector multiplication performance. The performance we take, proves that the computational rate increases with the degree of the polynomial.

![Figure 1: Polynomial Preconditioned Conjugate Gradient performance on the CM-2. \( C = 4 \) and \( n = 128K \).](image)

### 4 Conclusion

We show that sparse computations are possible on massively parallel architectures. We assume in our method that \( Cn \geq p \) to have a \( \text{wp} \geq 1 \) but we do not need \( n \geq p \) (where \( p \) is the number of physical processors). Problems from fluid mechanics, quantum chemistry or biology generate such problems. For dense or sparse computations, parallel machines will become more and more powerful. Teraflops peak performance and giga-word memories will soon allow us to access a new area of scientific computing. The challenge will be to propose efficient parallel algorithms for these machines. The researches presented in this paper are a step in this direction.

### References


PARALLEL IMPLEMENTATIONS FOR SOLVING GENERALIZED EIGENVALUE PROBLEMS ARISING FROM STRUCTURAL MECHANICS

Bernard PHILIPPE and Brigitte VITAL
IRISA - Campus de Beaulieu
35042 Rennes Cedex, FRANCE

Abstract - We compare three algorithms to solve the sparse symmetric generalized eigenvalue problem on vector and parallel computers. An efficient vectorization for tridiagonalizing a banded matrix is defined. Finally, we propose a decision tree for the procedure selection.

1. INTRODUCTION

The purpose here is to solve on a parallel computer the following generalized eigenvalue problem:

\[ Ax = \lambda Mx, \]

where \( A \) and \( M \) real symmetric matrices of order \( n \), such arise in Structural Mechanics. The assumptions imply real eigenvalues and the existence of an orthogonal basis of eigenvectors. The matrices \( A \) and \( M \), which are usually obtained from a discretization, are assumed to be large and sparse. Typically, their bandwidth does not exceed \( n/10 \). In fact, \( M \) is either diagonal or with the same nonzero pattern as \( A \). For non-diagonal matrices, we consider three possible storage techniques: banded matrix, profiled matrix or sparse matrix with an unstructured pattern. In all these situations, symmetric matrices are only half-stored. Since the large size of the matrices prevent the computation of the entire spectrum, the user must specify which part of the spectrum he is willing to compute. He may specify it either by limiting the interval of the eigenvalues sought or by looking for a given number of extremal eigenvalues.

The algorithms considered here are designed for parallel computers with a shared common memory such as the CRAY or ALLIANT computers. Three algorithms are developed depending on the type of storage used for the matrices. After a description of each of them, we compare their performance on some test problems and draw some conclusions about the method of choice for each situation.

All the above problems are defined on matrices which belong to the HARWELL set of test matrices. Since profile minimization may be done at a very small cost, the profile of the matrices was minimized in all cases.

2. REDUCTION TO A STANDARD EIGENVALUE PROBLEM

\( A \) : banded matrix; \( M \) : diagonal matrix

2.1. Analysis of the sequential algorithm

Since \( M \) is a positive definite matrix, the problem defined by Equation (1) is equivalent to the following standard eigenvalue problem:

\[ Cx = \lambda y, \quad x = Dy, \]

where \( D = M^{-1/2} \) and \( C = DA D \). Since in this section \( A \) is a banded matrix and \( M \) is diagonal, the matrix \( C \) is easily computed. It has the same band as the original matrix. Therefore, given an interval \([a,b]\), all the eigenvalues of the original problem which belong to the interval and their corresponding eigenvectors can be computed by the following algorithm:

1. Compute the matrix \( C \) defined in Equation (2).
2. Compute a tridiagonal matrix \( T \) unitarily similar to \( C \) using Givens transformations.
3. Compute all the eigenvalues of \( T \) which belong to \([a,b]\) by iterative bisections of the interval.

4. For each eigenvalue \( \lambda \), compute the corresponding eigenvector by inverse iterations (this implies the factorization of the matrix \( C = \lambda I \)).

5. M-orthogonalize groups of eigenvectors corresponding to close eigenvalues by using the Modified Gram-Schmidt procedure [2].

The algorithms of Steps 2 to 4 may be found in [5]. They have been implemented in the subset of the routines which are devoted to banded matrices in the EISPACK library.

Let us consider each stage with respect to parallelism and operation counts. The half-bandwidth of \( A \) is denoted \( p \).

Step 1 is easy to parallelize since it only involves vector operations (componentwise vector multiplications with vector length \( p \)). Moreover, its complexity is low \((2np)\) operations.

In Step 2, the tridiagonalization process involves vector operations but with vector length decreasing from \( p \) to 2 during the procedure with a \( O(n^2p) \) complexity for scalar operations. The algorithm is based on a sequence of rotations which gives rise to recurrences. It is essentially sequential.

A parallel algorithm for Step 3 is studied in [3]. Its efficiency is usually almost optimal but the low complexity of the stop limits its impact on the overall process.

Step 4 is intrinsically parallel since the inverse iterations may be independently performed.

In Step 5, two levels of parallelism may be exploited: between different groups of vectors and within the orthogonalization of one group. The implementation is studied in [4]. To orthogonalize a group of \( m \) vectors, \( 2m^2n \) operations must be performed.

In conclusion, Step 2 appears to be the bottleneck for parallelizing the whole process. We propose here a new version of the algorithm which allows efficient vectorization. Theoretically, whenever vectorization is possible, parallelizing should also be possible. However, on the computers considered here, we were not able to obtain adequate speed-ups. Hence, we restrict ourselves to the use of one vector processor.

2.2. Vectorizing the tridiagonalization of a banded matrix

The successive eliminations of entries of the banded matrix \( C \) are ordered as follows: diagonal by diagonal, starting from the outermost one, and on one diagonal, from top to bottom. Therefore, the computation is structured by a two-level loop:

\[ \text{do } q = p, p - 1, \ldots, 2 \]
\[ \text{do } i = 1, 2, \ldots, n - k \]

\[ \text{elimination of } A(i, i + q) \]

where elimination of \( A(i, i + q) \) involves a sequence of \( t = \lfloor \frac{n-k}{q} \rfloor \). Given rotations \( R(k, l, 0) \) that are successively applied to \( A \):

\[ A := R(k, l, 0)^T A R(k, l, 0), \quad k = 0, \ldots, l - 1. \]

The sequence of rotations comes from the fact that eliminating an entry gives rise to a new nonzero entry lower down on the side of the band. This new entry will have to be eliminated later. Therefore, elimination of \( A(i, i + q) \) is implemented by a two-level loop:

\[ \text{do } k = 0, 1, \ldots, l - 1 \]
\[ \text{computation of } R(k, l, 0) \]

\[ \text{loop for computing } A := A R(k, l, 0) \]
\[ \text{loop for computing } A := R(k, l, 0)^T A \]

At the beginning of the process, the vectorized inner loops of zero entries are efficient if the bandwidth is large. However, when eliminating entries close to the main diagonal, the number of rotations is high and at the same time the vectors being manipulated are very short. In this case, a vectorizer will not yield efficient code. This problem can be rectified by computing the inner and outer loops. This is permitted when the computation of all the rotations \( R(k, l, 0) \) may be taken out of body of the loop. This can be done by a recursion which computes the sines and cosines of the rotations [4].

We compared these two methods of vectorization with a third combined method. For every pair \((q, t)\), the implementation selects the
best strategy between the original one (strategy A) or the strategy which is obtained by permuting the loops (strategy B). The efficiency of the approach is described in Table 1.

<table>
<thead>
<tr>
<th>s, vector speed-up</th>
<th>7</th>
<th>50</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only (A)</td>
<td>s = 0.88</td>
<td>s = 1.51</td>
<td>s = 1.87</td>
</tr>
<tr>
<td>Only (B)</td>
<td>s = 3.20</td>
<td>s = 1.85</td>
<td>s = 1.27</td>
</tr>
<tr>
<td>switch between (A) and (B)</td>
<td>s = 2.93</td>
<td>s = 2.49</td>
<td>s = 2.49</td>
</tr>
</tbody>
</table>

Table 1: Vectorization of the tridiagonalization on Cray 2 (n=500, 1 CPU)

3. MULTISECTIONNING WITH STURM SEQUENCES

(A: profiled matrix, M: diagonal or profiled matrix)

3.1. Presentation of the algorithm
When A and M are profiled matrices with the same profile, the problem defined by Equation (1) can be solved by iteratively partitioning the interval [a, b] in which the eigenvalues are sought. Unlike the case in the previous section, the Sturm sequences are computed from the original matrices. The calculation implies $L - D - L^T$ factorizations of matrices $A - M^2$ where $\lambda \in [a, b]$. A factorization involves vector operations with a vector length equal to the half-bandwidth $p$. Parallelism is obtained by computing several sequences concurrently.

For that purpose, we adopted a strategy similar to the strategy used in the tridiagonal situation [3], namely:

1. Isolate the eigenvalues by multisectons of order $n_{proc}$, the number of processors (a multiseciton is a synchronous computation of n proc Sturm sequences). This step results in a list of intervals enclosing single eigenvalues is obtained.
2. Extract the eigenvalues from their interval and compute the corresponding eigenvectors. Here, Sturm sequence computations are asynchronos. We selected ZERON as the root finding procedure [1]. Eigenvectors are computed via inverse iterations.
3. $M$-reorthogonalize the groups of vectors corresponding to close eigenvalues.

3.2. Parallel implementation
We illustrate the behavior of the algorithm on a problem of order $n = 817$ and where the half-bandwith, $h$, of $A$ is equal to 18 but where $M$ is diagonal. The number of eigenpairs sought is 16. The speed-ups on ALLIANT FX/80 are displayed in Table 2. For 8 processors, the proportion of time spent in Step 1, Step 2 and Step 3 are respectively 12.3%, 87.0% and 0.6% of the total time.

<table>
<thead>
<tr>
<th>nb. processors</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>speed-up</td>
<td>1</td>
<td>2</td>
<td>3.6</td>
<td>6.6</td>
</tr>
</tbody>
</table>

Table 2: Multisectons with profiled matrices on an ALLIANT FX/80 (Total time on one processor: 234.6 seconds)

4. METHOD OF LANCZOS
(A: unstructured sparse matrix, M: diagonal or profiled matrix: extremal eigenvalues)

For solving Problem (1), we consider the Lanczos method with the following characteristics: block version with full reorthogonalization and dynamic restarting process. The algorithm is described in [4]. The iterative process builds an $M$-orthogonal system of vectors $V_j$ and a banded matrix $T_j = V_j^T A V_j$ at every step. At every iteration, the following steps are performed:

- premultiplication of a block by A;
- premultiplication of a block by $M$;
- dense computations between blocks of vectors;
- premultiplication of a block by $M^{-1}$;
- tridiagonalization of $T$ and computation of the eigenpairs (not at each iteration);
- $M$-reorthogonalization.

Use of blocks provides an obvious way to parallelize the algorithm by performing the transformations on the columns of a block independently. The tridiagonalization was not parallelized but only vectorized as seen in Section 2. On a problem of order $n = 1173$ where $A$ stored by sparse diagonals and $M$ is a profiled matrix ($p = 62$), the program was run on 4 processors of a CRAY 2. The time required to compute the 20 largest eigenpairs was 31.89 seconds (block size: $l = 20$). The corresponding speed-up was 2.9.

5. COMPARISONS AND CONCLUSION

The domain of applicability of each of the three presented algorithms is different, depending on the position of in the spectrum of the eigenvalues sought and on the matrix $M$. They have also different memory requirements. The highest requirement comes from the Multisectons which require the storage of one profiled matrix per processor. However this is also the algorithm which yields to the highest speed ups. To compare the methods, we present running times for two different problems in Table 3:

- Problem (I): $n = 1224$, $p = 56$, $M$ is not diagonal.
- Problem (II): $n = 1083$, $p = 62$, $M$ is diagonal.

It is clear that, when applicable, Lanczos method is the method of choice. The decision tree for the algorithm selection is given in Figure 1.

![Figure 1: Decision tree for algorithm selection.](image)

References

THE LANCZOS ALGORITHM FOR THE GENERALIZED SYMMETRIC EIGENPROBLEM ON SHARED-MEMORY ARCHITECTURES

Mark T. Jones
Mathematics and Computer Science Division Argonne National Laboratory 9700 South Cass Avenue Argonne, IL 60439

and

Merrell L. Patrick
Computer Science Department Duke University Durham, NC 27706

Abstract. The generalized eigenvalue problem, \( Kz = \lambda Mz \), is of significant practical importance, for example, in structural engineering where it arises as the vibration and buckling problems. The paper describes the implementation of a solver based on the Lanczos algorithm, LANZ, on two shared-memory architectures, the CRAY Y-MP and Encore Multimax. Issues arising from implementing linear algebra operations on a multivector processor are examined. Portability between a multivector processor and a simple multiprocessor is discussed. Performance results from some practical problems are given and analyzed.

1. INTRODUCTION

The generalized symmetric eigenvalue problem, \( Kz = \lambda Mz \), is of significant practical importance, for example, in structural engineering where it arises as the vibration and buckling problems. In the problems of interest, a few of the eigenpair closest to some point, \( \sigma \), in the eigenspectrum are sought. The matrices, \( K \) and \( M \), are usually sparse or have a narrow bandwidth. Because eigenvalue problems arising in structural engineering are often very large, it is natural to attempt to use parallel computers to solve them. In Section 2 the parallel LANZ algorithm and its implementation on shared-memory architectures with a small to moderate number of processors is described. In Section 3 results from the implementations are given and analyzed.

2. ALGORITHM AND IMPLEMENTATION

To speed convergence to desired eigenvalues, a shift-and-invert Lanczos algorithm similar to that described in [8] is used. On sequential and vector machines, this algorithm has been observed to be superior to the subspace iteration method that is popular in engineering [9] [2]. To maintain the desired semi-orthogonality among the Lanczos vectors, a version of partial reorthogonalization [14] is used. Extended local orthogonality among the Lanczos vectors is also enforced [7] [12]. If eigenvectors are found before executing the Lanczos algorithm, an improved version of external selective orthogonalization [1] suggested in [2] is used to avoid recomputing these eigenvectors. Although the discussions in this paper assume that \( M \) is positive semi-definite, the computations remain essentially the same when \( M \) is indefinite.

The Force, a Fortran-based language for parallel programming [5], was used to implement LANZ because it is available on several shared memory architectures, thus allowing at least a superficial level of portability.

The parallel LANZ algorithm is presented in Figure 1. Its various computational components and their parallel implementations are discussed in the following subsections. Explicit global synchronization points in the algorithm are denoted by the term "Synchronize." Other synchronization points are required by particular operations, for example inner products, and are not explicitly denoted in the algorithm. To avoid extra synchronization, each processor is responsible for computing a fixed subset of each vector computation. For example, if at step 21 processor \( i \) computes the first \( m \) elements of \( q_{i+1} \), then at step 22, processor \( i \) would compute the contribution of the first \( m \) elements to the inner product, thus avoiding a synchronization between steps 21 and 22. In these discussions \( p \) represents the number of processors, \( n \) represents the order of the matrices, \( b \) represents the block size in a block algorithm, and \( j \) represents the current Lanczos step.

![Fig. 1. Parallel shift-and-invert Lanczos algorithm](image-url)

2.1. Factorization

Factorization takes place only once during the algorithm, at step 7. Because the matrices, \( K \) and \( M \), are sparse (or have been reordered to have a narrow bandwidth), the parallel implementation of direct factorization and solution methods must be carefully considered. In this paper, only the case in which the matrices have been reordered to a narrow bandwidth, \( \beta \), will be considered. However, the limitations on parallelism in factorization and forward/backward matrix solution that are imposed by a narrow bandwidth are similar to those imposed by sparse matrices.

Two situations may exist when factoring \( (K - \sigma M) \). (1) \( (K - \sigma M) \) is known to be positive definite, and therefore it is desirable to use either Cholesky factorization or LDL^T decomposition, or (2) \( (K - \sigma M) \) may be indefinite, and therefore a factorization algorithm with pivoting is necessary. In the first case, a block factorization and
solution subroutine described in [13] has been parallelized for use in LANZ. In the second case, a block algorithm for banded matrices based on Bunch-Kaufman factorization is used [3].

LANZ was initially written for vector architectures, and therefore careful attention has been paid to achieving good vectorization. With small-to-moderate vector lengths, it is desirable to perform \textit{saxpy} operations\(^1\) as opposed to inner products, as well as to compute more than one \textit{saxpy} operation at a time.\(^2\) On multivector processors, however, good vectorization is often at odds with parallelization. In the factorization algorithms, this conflict between vectorization and parallelization occurs in the computation of the pivot column(s): the pivot column(s), vectors of length \(\beta\), must be split into vectors of length \(\beta/p\) for each processor to compute. On the CRAY Y-MP, the benefit of parallel computation of the pivot column is outweighed by the resulting inefficient short vector operations and the cost of the added synchronization; therefore, this computation is not parallelized. However, on a simple multiprocessor such as the Multimax, this conflict does not occur, and the computation of the pivot column is parallelized. The dominant part of the calculation is the updating of the uneliminated nonzeros by using the pivot columns: the updating is implemented by distributing \(\beta/p\) extended \textit{saxpy}'s to each of the processors to compute. The extended \textit{saxpy}'s parallelize well because there is sufficient work for each processor, and the vector lengths are unaffected by parallelization.

2.2. Matrix Solution

Forward and backward matrix solution is required at steps 7 and 16. The conflict between vectorization and parallelism is much worse in these operations. This discussion will be limited to the forward and backward solution algorithms that take place after a Bunch-Kaufman factorization in which the block sizes vary.\(^3\) are selected according to numerical criteria rather than the number of processors.\(^3\) The following discussion will assume that the lower triangular factor, \(L\), resulting from the Bunch-Kaufman algorithm has been stored by row.\(^4\) Because of the order in which pivots are performed, a \textit{saxpy}-based algorithm for the forward solution must be used, and an inner product algorithm for the backward solution must be used.

The time-consuming portion of the block forward solution algorithm is the \(b\ \beta\)-length \textit{saxpy} operations that can be combined into a single extended \textit{saxpy} operation. The only practical way to parallelize this operation is to split the vector into \(p\) shorter vectors. This approach, of course, significantly reduces the efficiency of the vector operations.

The time-consuming portion of the block backward solution algorithm is the computation of \(b\ \beta\)-length inner products. Two types of parallelism are available here. (1) Two or more processors can cooperate to compute a single inner product, and (2) individual inner products can be computed independently. Even though both methods are used, the algorithm is still inefficient because inner products are not as fast as \textit{saxpy}'s, the parallel computation of a short inner product is adversely affected by synchronization delays, and the block size may not be evenly divisible by \(p\), and therefore a load imbalance may result.

The considerations regarding efficiency of vector operations are not a concern when implementing this algorithm on the Encore, and therefore better parallel speedup from the forward and backward solution algorithms can be expected than on the CRAY Y-MP. The ratio of computation to synchronization, however, is still much worse than for factorization, and good speedup cannot be expected.

2.3. Other Computations

The parallelization of other computations in the algorithm, including sparse matrix multiplication, solution of the small tridiagonal eigenproblem, \textit{Ritz} vector computation, and orthogonalization and discussed more fully in [4].

![Speedup curves](image)

3. PERFORMANCE RESULTS AND ANALYSIS

As a demonstration of its performance, LANZ was run on a medium size eigenproblem from structural engineering\(^5\) where the ten lowest eigenpairs were found in 22 steps on a four processor CRAY Y-MP. A smaller problem\(^6\) was run on a twenty processor Encore Multimax in which the ten lowest eigenpairs were found in 22 steps. It is clear from the speedup curves in Figure 2 that a speedup plateau occurs. The main cause of this plateau is the poor speedup realized in the forward and backward matrix solution algorithms. The problem caused by the matrix solution algorithms is exacerbated as the number of Lanczos steps increases, because each Lanczos step requires another forward/backward matrix solution, taking more and more time as compared to factorization, which speeds up well. This plateau occurs later on the Encore than the CRAY because the Encore does not have to contend with the conflict between vectorization and parallelization—vector lengths decreasing as the number of processors increases. However, both implementations suffer from the poor ratio of computation to synchronization in the forward and backward matrix solution algorithms.

If problems with larger bandwidths were used, better speedup from these algorithms could, of course, be expected. It has been the authors' experience, however, that if the bandwidth arising from a structural engineering problem is large, then most likely many zeroes exist inside the band, and therefore sparse methods are best used.

---

\(^1\) The \textit{saxpy} operation is defined as \(w = w + y\), where \(w\), \(x\), and \(y\) are vectors and \(a\) is a scalar.

\(^2\) Performing more than one \textit{saxpy} at a time, called an extended \textit{saxpy} in this paper, is defined as \(w = w + \sum_{i=1}^{n} y_i\), and is often implemented via loop unrolling. This type of operation reduces the ratio of memory references to computations.

\(^3\) The situation is slightly better for the positive definite case in which the block sizes can be selected based on the number of processors rather than according to numerical criteria.

\(^4\) If it were stored by column, the same limitations would apply, but the discussion for the forward solution would be applicable to backward solution and vice versa.

---

\(^5\) Finding the vibration modes and mode shapes of the finite elements model of a circular cylindrical shell [15]. In this problem \(n = 13654\) and the average semi-bandwidth is 294.

\(^6\) Finding the five lowest buckling modes and mode shapes of the finite elements model of an I stiffened panel. In this problem \(n = 4474\) and the average semi-bandwidth was 207.
4. CONCLUDING REMARKS

A parallel-Lanczos algorithm for finding a few of the eigenpairs around a point in the eigenspectrum was described. Differences in the implementation of the algorithm on a multivector processor and on a multiprocessor were described. The algorithm was shown to perform reasonably well on a moderate number of processors. A performance bottleneck which prevents efficient utilization of a large number of processors was identified.

Several possible modifications to the LANZ algorithm can be used to improve its parallel performance. The use of dynamic shifting [1] to improve parallelism by reducing the number of forward and backward matrix solutions was investigated in [2] and was found to be successful when the eigenvalue distribution was difficult. The use of groups of processors executing the LANZ algorithm independently at different shifts was investigated in [3] and was found to be successful when many eigenpairs are being sought. Block Lanczos holds some promise because it allows several forward and backward matrix solutions to occur simultaneously [1]. The improvement in performance resulting from block Lanczos will depend on how many Lanczos steps are eliminated and what block size can be effectively used. Unfortunately, s-step Lanczos methods [4] will not alleviate the bottleneck imposed by forward and back solutions and, therefore, will not have a significant effect on performance.

Another avenue for improving parallel performance is the use of iterative methods to solve \((K - \sigma M)z = y\) rather than direct methods. However, it has been the authors' experience that \((K - \sigma M)\) is often poorly conditioned and, therefore, is difficult to solve by iterative methods.

REFERENCES


A DIVIDE AND CONQUER APPROACH TO THE NONSYMMETRIC EIGENVALUE PROBLEM

ELIZABETH R. JESSUP  
Mathematical Sciences Section  
Oak Ridge National Laboratory  
P.O. Box 2008, Building 6012  
Oak Ridge, TN 37831-6367 USA

Abstract Serial computation combined with high communication costs on distributed-memory multiprocessors make parallel implementations of the QR method for the nonsymmetric eigenvalue problem inefficient. This paper introduces an alternative algorithm for the nonsymmetric tridiagonal eigenvalue problem based on rank two tearing and updating of the matrix. The parallelism of this divide and conquer approach stems from independent solution of the updating problems.

I. Introduction

The eigenvalues and eigenvectors of a nonsymmetric matrix $A$ have traditionally been computed by first reducing $A$ to Hessenberg form $H$ and then computing the eigendecomposition of $H$ by the QR method. The serial nature of the QR method combined with the high cost of data transfer on distributed-memory multiprocessors has made parallel implementations of this approach inefficient [7]. In this paper, we outline an alternative algorithm for the nonsymmetric eigenvalue problem. The algorithm uses a divide and conquer technique and follows from methods that have performed well both serially and in parallel for the symmetric tridiagonal [4] and tridiagonal [2] and unitary [2] eigenvalue problems and for the bidiagonal singular value problem [10].

The method is presented here for nonsymmetric tridiagonal matrices. Matrices of this form arise directly from certain applications [1] and from reduction of general matrices to tridiagonal form [5]. We expect ultimately to extend our divide and conquer technique to general matrices.

Throughout this paper, unless otherwise specified, capital Greek and Roman letters represent matrices, lower case Roman letters represent column vectors, and lower case Greek letters represent scalars. A superscript $T$ denotes transpose. The vector $e_j$ is the $j$-th "canonical vector" with all elements equal to zero except the $j$-th which equals 1.

II. The Algorithm

Let $T$ be the following $n \times n$ real, tridiagonal, irreducible, nondefective nonsymmetric matrix

$$ T = \begin{pmatrix} \alpha_1 & \beta_1 & & & & \\ \gamma_2 & \alpha_2 & \beta_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \gamma_{n-1} & \alpha_{n-1} & \beta_{n} & \\ & & & \gamma_n & \alpha_n & \beta_{n+1} \end{pmatrix}. \quad (1) $$

By splitting off two superdiagonal elements $\beta_m$ and $\beta_{m+1}$ and the corresponding subdiagonal elements $\gamma_m$ and $\gamma_{m+1}$, we can write the matrix $T$ in terms of the tridiagonal nonsymmetric submatrices $T_1$ and $T_2$:

$$ T = \begin{pmatrix} T_1 & \beta_m^T \\ \gamma_m T_2 & \beta_{m+1} \end{pmatrix}. \quad (2) $$

If $T_1$ and $T_2$ are nondefective, we can compute the eigendecompositions $T_1 = X_1 D_1 X_1^{-T}$ and $T_2 = X_2 D_2 X_2^{-T}$. Substituting these decompositions and abbreviating $\alpha = \alpha_m$ gives the matrix product

$$ T = \begin{pmatrix} X_1 D_1 X_1^{-T} & \beta_m^T \\ \gamma_m T_2 & \beta_{m+1} \end{pmatrix} \equiv \hat{X} \begin{pmatrix} D_1 & \beta_m^T \\ \gamma_m T_2 & \beta_{m+1} \end{pmatrix} \hat{X}^{-1}, \quad (3) $$

where

$$ \hat{X} = \begin{pmatrix} X_1 & 1 \\ -X_2 & 1 \end{pmatrix}, $$

$v_1 = \beta_m^{-1} c_m$, $v_2 = \gamma_{m+1} X_2^{-1} c_1$, $h_1 = \gamma_m X_1^{-1} c_m$, and $h_2 = \beta_{m+1} X_2^{-T} c_1$ for canonical vectors $c_1$ and $c_m$ of appropriate length. (The algorithm can be reformulated to account for defective matrices by replacing the eigendecompositions $T_1 = X_1 D_1 X_1^{-T}$ and $T_2 = X_2 D_2 X_2^{-T}$ with ones including the rank deficient left and right eigenvector matrices $T_1 = X_1 D_1 Y_1$ and $T_2 = X_2 D_2 Y_2$.)

We permute the elements of equation (3) to form

$$ T = \hat{X} \begin{pmatrix} D_1 & \beta_m^T \\ \gamma_m T_2 & \beta_{m+1} \end{pmatrix} \hat{X}^{-1} \quad (4) $$

with

$$ \hat{X} = \begin{pmatrix} X_1^{-T} & 1 \\ -X_2^{-T} & 1 \end{pmatrix}, $$

and rewrite the interior matrix of equation (4) as

$$ M = \begin{pmatrix} D_1 & \beta_m \beta_m^T \\ \gamma_m T_2 & \beta_{m+1} \beta_{m+1}^T \end{pmatrix} \begin{pmatrix} v_1 & v_2 \\ h_1 & h_2 \end{pmatrix}. \quad (5) $$

The matrix $M$ is the sum of a diagonal matrix and a rank-two nonsymmetric matrix. The eigenvalues of the matrix $M$ are the eigenvalues of $T$. The left and right eigenvectors of $M$ postmultiplied by $X_T$ and $X^{-1}$, respectively, are the left and right eigenvectors of $T$.

The procedure for computing the eigenvalues and eigenvectors of $M$ follows basic steps similar to those for the eigendecomposi-
tion of a diagonal plus symmetric rank one matrix developed in [3,8], but because \( M \) is nonsymmetric, the details differ in several important ways. Let \( \delta_1, \ldots, \delta_n \) denote the diagonal elements of \( D \). If no eigenvalue \( \lambda \) of \( M \) equals a diagonal element \( \delta_j \), then the eigenvalues of \( M \) are the zeros of the rational equation

\[
g(\lambda) = (\lambda - \alpha) + \sum_{i,j=1}^{n-1} (c_i \bar{c}_j \delta_i \delta_j) = 0
\]

which may be solved using a complex rootfinder such as Muller's method [11]. Because they are also the eigenvalues of a submatrix of the real tridiagonal matrix \( T \), the complex eigenvalues of \( M \) occur in conjugate pairs.

Once the eigenvalues have been computed, it is a straightforward matter to compute the eigenvectors using the expression for the matrix \( M \). Let \( \lambda \) be a computed eigenvalue of \( M \). One expression for the corresponding right eigenvector \( q^T = (q_i, \xi) \) comes from the relation

\[
(D - \lambda) (q_i, \xi) = (Dq_i + \xi \epsilon) = \lambda (q_i, \xi).
\]

Specifically, if the matrix \( (D - \lambda) \) is nonsingular, the right eigenvector is given by

\[
q = (q_i, \xi) = \frac{(-D - \lambda)^{-1} \epsilon}{1} \xi,
\]

where \( \xi \) is selected to make \( q^T q = 1 \). A similar expression can be derived for the corresponding left-singular vector.

To this point, the derivation of the divide and conquer method for the nonsymmetric eigenvalue problem requires that the matrix \( (D - \lambda) \) be nonsingular. For this to be the case, no eigenvalue of \( M \) may equal a diagonal element of \( D \). Equivalently, no element of \( u \) can be zero, no element of \( h \) can be zero, and no diagonal elements of \( D \) may be equal. We now show how to deflate the problem in exact arithmetic so that none of these situations arises. The deflation rules not only produce an exact correct form for the matrix \( M \) but also reduce the amount of computation needed for solving its eigensystem.

First, if the \( j \)th element of \( v \) is zero, then \( c_j^T M = \lambda c_j^T v \) where \( c_j \) is the \( j \)th canonical vector of length \( n \). Thus, \( \lambda_j \) is an eigenvalue of \( M \) and \( c_j \) is its corresponding left eigenvector. Likewise, if the \( j \)th element of \( h \) is zero, then \( M c_j = \lambda_c c_j \), so that \( (\lambda, c_j) \) is a right eigenpair of \( M \). Rows and columns of the matrix \( M \) having all zero off-diagonal elements can be removed from the matrix and the eigenvalue problem deflated.

When \( \xi = \delta_j \), we can apply unitary similarity transformations to reduce \( h_i = c_i^T h \) to zero and deflate the problem. Let

\[
r^2 = |c_j|^2 + |h_j|^2, \quad \epsilon = \frac{h_j}{r}, \quad s = \frac{c_j}{r},
\]

then the matrix is transformed in the following way

\[
\begin{pmatrix}
\epsilon & -s \\
s & \epsilon \\
0 & 1
\end{pmatrix}
\begin{pmatrix}
\delta_j & 0 \\
0 & \delta_j \\
0 & 0
\end{pmatrix}
\begin{pmatrix}
\epsilon & s \\
-s & \epsilon \\
0 & 0
\end{pmatrix}.
\]

After transformation, \( \delta_j \) is an eigenvalue of \( M \) with right eigenvector \( \epsilon, s \). Note that the zero structure of the deflated matrices guarantees that the \( n \times n \) matrix \( M - \lambda I \) has rank at least \( n - 1 \). Thus, \( M \) has distinct eigenvalues and a complete set of eigenvectors.

As in the symmetric case, we expect the divide and conquer method to be a good parallel method. The rank two tearing of equation (2) can be applied recursively to tridiagonal submatrices \( T_1 \) and \( T_2 \). Parallelism arises both at the subproblem level (computing \( T_1 = X_1 D_1 X_1^{-1} \) and \( T_2 = X_2 D_2 X_2^{-1} \) in parallel) and at the rootfinding level. Both levels can be exploited in shared-memory implementations of divide and conquer algorithms [4,10], but no implementation on a distributed-memory multiprocessor has attempted to parallelize the rootfinding tasks [9]. As in the symmetric case, the nonsymmetric divide and conquer algorithm can be pipelined with reduction of a general matrix to tridiagonal form by the algorithm in [6]. As soon as a submatrix has been formed, its eigendecomposition is computed. Once additional subproblems have been solved, rank one updating can begin.

References


A PARALLEL VARIANT OF GMRES(m)

ERIC DE STURLER
Delft University of Technology
Faculty of Technical Mathematics and Informatics
P.O. box 356, NL-2600 AJ Delft, The Netherlands

Abstract: In the usual implementation of GMRES(m) [3] the computationally most expensive part is the Modified Gram-Schmidt process (MGS). It is obvious, that the MGS process is not well parallelizable on distributed memory multiprocessors, since the inner products act as synchronization points and thus require communication that cannot be overlapped. Furthermore, as all orthogonalizations must be done sequentially, MGS generates a large number of short messages, which is relatively expensive. Especially on large processor grids the time spent in communication in the MGS process may be significant. For this reason a variant of the usual GMRES(m) algorithm is considered, called modGMRES(m), which first generates the vectors that span the Krylov space and then combines the MGS steps for a group of vectors. It is shown, on real world problems, that the modGMRES(m) method can yield a considerable gain in time per iteration. Numerical experience suggests that the total number of iterations remains about the same as for GMRES(m).

Introduction

As described in [2] at SHELL’s KSEPL the reservoir simulator Bosim has been parallelized on a Meiko Computing Surface, a transputer based parallel computer. The parallelization is based on a 2-D domain decomposition, where the reservoir is divided among a 2-D grid of processors, in addition there is a master processor, which handles the initialization and input/output. In Parallel Bosim the largest part of the computation is done in Fortran. On each processor, the Fortran program runs in parallel with an Occam process. The Occam processes on different processors form a harness that takes care of the communication. When Fortran on a processor has to communicate, it sends the data to the local Occam process and then continues until its next communication. Meanwhile, the Occam processes concurrently take care of the communication and see to it that the data is returned to Fortran on the receiving processor(s). In this way communication and computation can be overlapped. Within this Parallel Bosim the GMRES(m) and modGMRES(m) method were implemented and compared. In Bosim the convergence in the linear solver is checked by a separate routine, which only computes the residual, and consequently will result in an inaccurate representation of \(H_{m+1}\). Therefore the \(v_i\) might be generated as follows:

\[
\begin{align*}
\text{do } i = 1, m \\
\text{do } j = 1, m + 1 \\
\text{enddo} \\
\text{enddo}
\end{align*}
\]

In this implementation communication is mostly overlapped and some time is saved by combining small messages, corresponding to one group of orthogonalizations, in one large message, which saves startup times and Fortran-Occam communication.

The generation of the vectors, that span the Krylov space, can be handled in two ways. If the condition number of the preconditioned system is sufficiently small and \(m\) is also relatively small (e.g. 10 or 20), the vectors can be generated as \(v_1, A v_1, A^2 v_1, \ldots, A^m v_1\), where \(A\) is the preconditioned matrix; this will be referred to as version 1. However for larger \(m\) and/or a preconditioned matrix \(A\) with a large condition number, the matrix \(v_1, A v_1, \ldots, A^m v_1\) will be so poorly conditioned that orthogonalization with the MGS algorithm will not produce a set of sufficiently orthogonal \(v_i\) and consequently will result in an inaccurate representation of \(H_{m+1}\).

The modGMRES(m) method

Let \(A x = b\) be the preconditioned system, and let \(r = b - A x\) be the residual. Let \(v_1\) be the normalized residual, then from \(v_1\), suitable set of vectors \(v_2, \ldots, v_{m+1}\), which span the Krylov space, is generated (see below). Then in the MGS process these vectors are orthogonalized. Because the Krylov space is generated in a different way, the Hessenberg matrix \(H_{m+1} = V^T A V\) must be computed from the coefficients of the MGS process, where \(V\) is the matrix \([v_1, v_2, \ldots, v_{m+1}]\). The rest of the method is analogous to the GMRES(m) method. See however also [1].

In the MGS process the vectors \(v_1, v_2, \ldots, v_{m+1}\) are orthogonalized in the following steps:

1. orthogonalize \(v_2, \ldots, v_{m+1}\) on \(v_1\), normalize \(v_2\), which gives \(v_2\)
2. orthogonalize \(v_3, \ldots, v_{m+1}\) on \(v_2\), normalize \(v_3\), which gives \(v_3\)

Each step can be done blockwise, because the orthogonalizations in one step are all independent. As the innerproducts computed on each processor are strictly local, these local coefficients must be accumulated over the processor grid to compute the global coefficients. This is done blockwise by the routines \(\text{acce}(\text{array}, \text{len})\), which sends the array to Occam and returns so that Fortran can continue, while Occam performs the actual accumulation in parallel, and \(\text{accr}(\text{array}, \text{len})\), which receives the accumulated values from Occam. The MGS process is implemented as follows (locally on each processor):

\[
\begin{align*}
d & = 1, m \\
\text{do } k = 1, m + 1 \\
\text{do } j = 1, nbl \\
\text{enddo} \\
\text{enddo}
\end{align*}
\]

In the above implementation communication is mostly overlapped and some time is saved by combining small messages, corresponding to one group of orthogonalizations, in one large message, which saves startup times and Fortran-Occam communication.
The computational costs for modGMRES(m) version 1 on each processor are given by:

\[ C_{\text{res,m},1} = \frac{1}{2}(m^2 + 3m)T_{\text{daxpy}} + \frac{1}{2}(m^2 + 3m)T_{\text{dealloc}} + (m + 1)T_{\text{prec}} + (m + 1)T_{\text{lincon}} + T_{\text{hes}}(m) \]  
(2)

and for version 2 by:

\[ C_{\text{res,m},2} = \frac{1}{2}(m^2 + 3m)T_{\text{daxpy}} + \frac{1}{2}(m^2 + 5m)T_{\text{dealloc}} + (m + 1)T_{\text{prec}} + (m + 1)T_{\text{lincon}} + T_{\text{hes}}(m) \]  
(3)

where \( T_{\text{daxpy}} \) is the time overheard measured in the daxpy's and \( T_{\text{dealloc}} \) which overlap the accumulation, \( T_c \) is some constant time and \( T_{\text{lincon}} \) gives a time increase per number. \( T_{\text{hes}} \) indicates the cost of non-overlapped communication/accumulation.

### Results

The GMRES(m) and the modGMRES(m) method were compared simulating a real reservoir for a large number of iterations. As the simulation had to be done on a fairly small machine (1 master and 24 gridnodes), whereas the modGMRES(m) method will only be really advantageous on a relatively large processor grid, involving 100 or more processors, a small reservoir was simulated on a 1D (line) processor grid. This gives 24 communication steps for accumulating a distributed value over the grid, which compares to a processor grid of some 150 processors. The number of unknowns in this problem was 13500 (active) gridblocks, which gives a maximum of 105 unknowns on a processor. With an average of some 90 unknowns per processor and 150 processors this compares to a reservoir with about 13500 (active) gridblocks, which is a medium scale reservoir model. This produced the following average timings on the busiest processor:

<table>
<thead>
<tr>
<th>computational costs</th>
<th>time (ms)</th>
<th>communication costs</th>
<th>time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T_{\text{daxpy}} )</td>
<td>0.30594</td>
<td>( T_{\text{lincon}} )</td>
<td>0.62041</td>
</tr>
<tr>
<td>( T_{\text{dealloc}} )</td>
<td>0.24707</td>
<td>( T_{\text{dealloc}} )</td>
<td>0.16000</td>
</tr>
<tr>
<td>( T_{\text{dealloc}} )</td>
<td>0.12354</td>
<td>( T_{\text{dealloc}} )</td>
<td>0.06400</td>
</tr>
<tr>
<td>( T_{\text{dealloc}} )</td>
<td>2.8251</td>
<td>( T_c )</td>
<td>0.35625</td>
</tr>
<tr>
<td>( T_{\text{prec}} )</td>
<td>2.4320</td>
<td>( T_{\text{lincon}} )</td>
<td>0.16293</td>
</tr>
<tr>
<td>( T_{\text{lincon}} )</td>
<td>3.0336</td>
<td>( T_{\text{hes}} )</td>
<td>0.9064</td>
</tr>
<tr>
<td>( T_{\text{hes}}(10) )</td>
<td>1.2060</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( T_{\text{hes}}(20) )</td>
<td>7.4160</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( T_{\text{hes}}(50) )</td>
<td>96.046</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Inserting these timings in (1) and (4) for GMRES(m), in (2), (5) and (6) for modGMRES(m) version 1, for \( m = 10 \) or 20, and in (3), (5) and (6) for modGMRES(m) version 2, for \( m = 50 \), leads to the following tables:

<table>
<thead>
<tr>
<th>( m )</th>
<th>( \text{time (ms)} )</th>
<th>GMRES(m)</th>
<th>modGMRES(m)</th>
<th>time diff.</th>
<th>( % )</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>126.12</td>
<td>138.49</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>313.43</td>
<td>355.51</td>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>1390.2</td>
<td>1832.3</td>
<td>32</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

These theoretical efficiency figures are based upon the model described above, with 13500 grid blocks, a processor grid of 150 processors and one master processor. The number of communication steps for accumulating a distributed value is 24 and the maximum number of blocks on a processor is 105. Furthermore communication costs, other than in the global accumulation of distributed values, are negligible. For GMRES(10), GMRES(20), modGMRES(10) and modGMRES(20) the overall timings were also determined experimentally on the busiest processor, which leads to the following table, containing the average time per iteration:

<table>
<thead>
<tr>
<th>( m )</th>
<th>( \text{time (ms)} )</th>
<th>GMRES(m)</th>
<th>modGMRES(m)</th>
<th>time diff.</th>
<th>( % )</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>126.09</td>
<td>137.13</td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>314.58</td>
<td>383.94</td>
<td>22</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

From these tables it is obvious that the modGMRES(m) method can yield a substantial improvement in time per iteration. Furthermore, for these simulations, there was no difference in the total number of iterations between GMRES(m) and modGMRES(m). With respect to the (theoretical) parallel efficiency, it can be seen that, for increasing \( m \), the decrease in efficiency of modGMRES(m) is much less than for GMRES(m). This is explained by the fact, that the communication costs increase exponentially with \( m \), and these are much higher for GMRES(m) than for modGMRES(m).

### References


PERFORMANCE OF ITERATIVE METHODS FOR DISTRIBUTED MEMORY MACHINES

D.C. Marinescu, I.R. Rice and E.A. Vavalis
Purdue University
Computer Science Department
West Lafayette, IN 47907, U.S.A.

Abstract: We consider the performance of the Jacobi SI iterative method applied to linear systems arising from partial differential equation problems. The E/T methodology is applied to experimental data from a 128 processor NCUBE 1. The observed performance of the implementation is seen to be poor, reasons for this are presented and remedies suggested.

1. INTRODUCTION

We describe the applications of the Event Thread of Control (E/T) methodology for parallel performance evaluation to iterative methods implemented on an NCUBE 1 (a distributed memory machine). The E/T methodology is to collect traces of events and from their behavior infer certain aspects of program performance, see [2]. Several kinds of events are traced, of interest here are Read and Write events as they relate to the communication and synchronization delays that occur in parallel iterative methods.

The key item in the E/T methodology is the characteristic function \( g \) defined by \( E = g(P) \) where \( E \) is the number of events and \( P \) is the number of threads or processors. The methodology uses assumptions of conservation and monotonicity of work to derive various relationships among performance measures, e.g., speed up, work, load balancing. A typical result is: If \( g(P) \) is increasing and convex, then the work per thread, \( W(P) \), is convex. Let \( P^* \) be the unique solution of \( P = \frac{W(1)/\theta + g(P)g'(P)}{g(P)} \), then the speed up is a maximum at \( P^* \). Here \( \theta \) is the additional work required of a thread when an event occurs and is characteristic of the hardware and operating system.

The iteration considered is the Jacobi SI method applied to the linear system that arises from solving an elliptic PDE in two dimensions. This iteration is inherently parallel and can be summarized on a \( P \) processor machine as follows:

Initialize

For NITER = 1 to LAST

For NPROC = 1 to P

Do all iterations on my equations

Send my boundary variables to neighbors

Receive neighbors boundary variables

End

End

The actual computation is complicated by steps to test for convergence and to adapt iteration parameters. These steps require global communication.

The NCUBE 1 used has 128 processors with both communication and computation handled by the single processor at each node. Communication is expensive on this machine relative to computation.

2. EXPERIMENTS AND MONITORING

The TRIPLEX tool set [1] is used to collect the trace data for two computations:

- **solid:** PDE problem with \( 33 \times 33 \) grid (about 1000 unknowns) on a non-rectangular domain (see Figure 5). Two to 128 processors used.
- **dash:** PDE problem with \( 50 \times 50 \) grid (about 2500 unknowns) on a square. Four to 64 processors used.

The names solid and dash refer to the lines used in the graphical data given below. The communication procedure used is to send all data from one processor to all other processors (broadcast). This inherently inefficient scheme was used because (a) the mapping of subdomains to processors was not known, (b) the system utilities for multicasting do not work properly, (c) we wanted to guarantee that the converges properties of the iterative method remain the same. Time is measured in units of a tick which is about 0.1 usec. Five iterations were made.

3. MEASUREMENTS AND DISCUSSION

Figure 1 shows the most basic data obtained in the E/T methodology, the characteristic function \( E = g(P) \), the number of events per thread (or processor). The average (expected) number of events per thread are shown as a function of \( P \) using solid and dash lines for the two experiments. The 95% level confidence intervals as computed from the experimental data are shown by + (for the solid case) and * (for the dash case). We see that \( g(P) \) grows linearly with slope about two on this semi-logarithmic plot so \( g(P) \approx O(P^2) \). It is known [2] that such a computation cannot exploit high parallelism well.

![Graph showing number of events per thread](image)

Figure 1: The expected number of events per thread of control and a 95% confidence interval for it.

Blocking is an important source of low performance, we illustrate the phenomena in Figure 2. At time \( t_1 \) processor \( P_i \) executes a READ and at time \( t_4 \), the requested result is available. The time \( t_2 - t_1 \) is **algorithmic blocking**, the time spent waiting because processor \( P_i \) has not yet computed the data \( P_i \) requests. The time \( t_6 - t_5 \) is **propagation delay**, the time spend waiting for the WRITE operation to be executed and the time \( t_4 - t_3 \) is **transmission delay**. The propagation and transmission delays on the NCUBE 1 are much larger than the ones in second and third generation hypercubes, but it is usually algorithmic blocking that causes very poor performance. The total time \( t_4 - t_1 \) is often called **synchronization delay**.

---

*This research supported in part by NSF grant CCR 86-12917 and by the Strategic Defense Initiative through ARO grants DAAH04-86-K-0109 and DAAH04-86-K-0107.
Figure 2: Communication involving blocking.

Figure 3 shows the expected algorithmic blocking time as a fraction of the total blocking time for these two PDE computations. The most obvious fact is that algorithmic blocking increases strongly as the number of processors increases. Note that the irregular behavior of these two curves are similar and can be explained from a detailed analysis of the characteristics of this computation [3].

Figure 4 shows the computing time as a fraction of the total non-blocked time per thread. Here time is partitioned into blocked (see Figure 2), performing I/O (Read or Write), and computing (presumably the useful work). We see that the fraction of computing time decreases rapidly with increasing P, in an ideal case one hopes for this to decrease much more slowly.

Figure 3: The expected algorithmic blocking time as a fraction of the total blocking time during a read operation.

4. CONCLUSIONS

Our E/T model predicts that parallelism cannot be effectively exploited if the characteristic function \( g(P) \) is quadratic in \( P \). Our experiments show that \( g(P) \) is quadratic and that the speed up obtained is low. Further analysis of this data [1] yields other conclusions, namely: (1) The cost of synchronization of the iteration is quite high, (2) The Jacobi SI code adapts the iteration parameters and tests for termination often, this results in even poorer performance due to the relative inefficiency of these computations for a distributed memory machine. A self-synchronization approach is presented and discussed in [2] which can alleviate these synchronization delay problems. Other approaches to improve performance are presented in [3].

Figure 4: The expected computing time as fraction of the non-blocked time per thread.

5. REFERENCES


685
Parallelizing ITPACK 2D

David R. Kincaid and Malathi Ramdas

Abstract. ITPACK 2D is a research-oriented numerical software package of iterative algorithms for solving large sparse systems of linear equations developed in the Center for Numerical Analysis at The University of Texas at Austin. The emphasis of this paper is to report on the project of parallelizing this software on the eight processor Cray Y-MP supercomputer using its advanced multiprocessing facilities. Model problems in two- and three-dimensions are used to test the performance of the parallelized version of the package using one to eight processors. Numerical results are used for comparing the performance of the routines in this package, for determining the speedup ratios for them, and for drawing general conclusions with regard to the efficiency and parallelizability of each of the methods.

I. Introduction. The ITPACK Project was started over a decade ago in the Center for Numerical Analysis of The University of Texas at Austin to conduct basic research on iterative algorithms for solving large sparse systems of linear algebraic equations. The emphasis has been on developing, testing, and evaluating software for solving linear systems arising from partial differential equations discretized using finite-difference and/or finite-element methods. Several ITPACK packages have been developed, modified, improved, and changed through various versions.

In the ITPACK 2D package, the basic iterative methods Jacobi (J) Reduced System (RS), and Symmetric Successive Overrelaxation (SSOR) are combined with two acceleration procedures; namely, Chebyshev (C) and Conjugate Gradient (CG). Also, included in the package is the Symmetric Successive Overrelaxation (SOR) method. The package uses adaptive procedures for the selection of the acceleration parameters and for automatic stopping tests. These routines work best when solving systems with symmetric positive definite or mildly nonsymmetric coefficient matrices.

The basic iterative routines available in the package are as follows.

- **JCG**: Jacobi Conjugate Gradient
- **JST**: Jacobi Semi-Iteration
- **SOR**: Successive Overrelaxation
- **SSORCG**: Symmetric SSOR Conjugate Gradient
- **SSORSI**: Symmetric SSOR Semi-Iteration
- **RS**: Reduced System Conjugate Gradient
- **RST**: Reduced System Semi-Iteration

Two ordering of the unknowns in the linear system are available: the natural (textigraphic) ordering and the red-black (checker-board) ordering.

II. Vectorization. ITPACK 2D [8, 10] is a modified version of ITPACK 2C [11] with enhanced vectorization capabilities. The primary changes made for the vectorized version [16] are:

1. Changing the storage format for the coefficient matrix from the "Yale sparse storage format" [8] to the "ELPACK matrix storage format" [15] since the latter is more vectorizable, and
2. Using a wavefront ordering (ordering by diagonals) to enable SOR, SSORCG, and SSORSI routines to vectorize under natural ordering.

The primary vectorizable and parallelizable operations in the iterative algorithms in ITPACK 2D [7] are matrix-vector multiplications, forward solves and backward solves.

A. Matrix-Vector Multiplication. The Yale sparse matrix format [8], as used in ITPACK 2C, is a row-wise sparse storage format using three linear arrays a, ja, ia. With this data structure, a matrix-vector multiplication of the form \(Au\) results in operations where the maximum vector length is equal to the number of nonzero elements in the row, which may be small for sparse matrices.

In the ELLPACK sparse matrix storage format [15], a rectangular array \(ccoef\) stores the nonzero elements of the scaled coefficient matrix \(A\) in a row-wise fashion. [The original linear system \(x = \beta\) is initially scaled by the diagonal \(D = \text{diag}(A)\) with \((D^{1/2}A D^{-1/2})D^{1/2}x = D^{1/2}\beta\) so that it assumes the form \((I - Cu) = \eta\). The entries in \(ccoef\) are also reordered when the red-black or wavefront ordering is used.] Another rectangular array \(jccoef\) stores the column numbers of the corresponding elements in \(ccoef\). If \(nnzmax\) is the maximum number of nonzero elements per row in \(A\) and \(n\) is the number of columns, then the matrix-vector product \(w = x + Gu\) can be computed with this data structure as follows.

```plaintext
do 20 j = 1,nnzmax
   do 10 i = 1,n
      w(i) = rhs(i)
   10 continue
   do 15 i = 1,n
      w(i) = w(i) + ccc(i,j)*u(jccoef(i,j))
   15 continue
20 continue
```

B. Forward and Backward Solves. Forward/Backward solves of sparse triangular matrices are a major component of the SOR and SSOR routines with the natural ordering. Since these routines do not vectorize in this situation, the unknowns are re-organized using a wavefront ordering. Several authors [1, 2, 3, 16] have noted that this ordering can effectively vectorize the forward solution process of a 5-point finite-difference stencil on a rectangular grid.

Although this concept of wavefront ordering is applied in ITPACK 2D to vectorize the SOR and SSOR routines in the case of natural ordering, the vector lengths still must be shorter than for the Jacobi or the Reduced System routines.

III. Microtasking. The basic parallelizing technique used is to partition do-loops into segments. This partitioning is done by introducing an outer loop that defines these partitions and assigns them to available processors. The beginning of a parallel do-loop is marked using a control command directive.

For example, the computation of a matrix-vector product \(w = x + Gu\) is shown below.

```plaintext
CHICS DO ALL SHARED (ntask,n,nnnzmax,nnz,rhs,ccoef,jccoef),
CHICS+ PRIVATE (i,j,ist,ied)
do 25 k = 1,ntask
   ist = (k-1)n/ntask + 1
   ied = k*n/ntask
   do 10 i = ist,ied
      u(i) = rhs(i)
   10 continue
   do 20 j = 1,nnzmax
      do 15 i = 1,ntask
         u(i) = u(i) + ccc(i,j)*u(jccoef(i,j))
      15 continue
20 continue
25 continue
```

Here \(ntask\) is the number of available processors. The \(CHICS\ DO ALL\) directive enables parallel execution using several processors with each computing a range of rows determined by the variables \(istan\) and \(ied\). For this directive, the scope of each variable (shared and private) used within the region must be defined. Notice that no synchronization is needed within the loops. Forward and backward solves are also parallelized using the same basic idea of partitioning the outer loop.

For the iterative algorithms under red-black ordering, the matrix-vector multiplication can be partitioned in such a way that all the red points are updated in parallel and then all the black points. Thus, a single synchronization point is needed. In the Reduced System method based on the black points, for example, the computation is \(w = x + F_{\text{red}}w\) and it can be carried out as shown in the displayed code below.

```plaintext
CHICS PARALLEL SHARED (ntask,nr,nb,nnnzmax,nnz,rhs,ccoef,jccoef),
CHICS+ PRIVATE (i,j,ist,ied)
do 25 k = 1,ntask
   ist = (k-1)nb/ntask + nr + 1
   ied = k*nb/ntask + nr
   do 10 i = ist,ied
      u(i) = rhs(i)
   10 continue
   do 20 j = 2,nnzmax
      do 15 i = 1,nr
         u(i) = u(i) + ccc(i,j)*u(jccoef(i,j))
      15 continue
20 continue
25 continue
```

CHICS END DO

686
The speedup ratios obtained for the larger problems (both 2-D and 3-D) were better than that obtained for the smaller problem. Under natural ordering, the JCG and JSI routines gave good speedup results while the SR and SSOR routines performed quite poorly. In general, the wavefront ordering used for these routines resulted in short vector lengths and also required the processors to synchronize too often. However, the SR and SSOR routines had improved speedups for the natural ordering for the 3-D problems. This was due to the fact that there were fewer wavefronts for the 3-D problems in comparison to the 2-D problems. This minimized the number of synchronization points. Under red-black ordering, all routines had good speedups. For both ordering, the speedup ratios obtained for the 3-D problems were not as good as those obtained for the 2-D problems.

Since the largest speedup does not imply the fastest procedure, the recommended routines in this parallel package are SSORC when the unknowns can be reordered into the red-black ordering and JCG otherwise.

Acknowledgements. The authors gratefully acknowledge the helpful advice and useful suggestions given to them by Tom Oppe, Irima Mukhejee, and David Young.

This work was accomplished with the aid of the Cray Y-MP supercomputer at Cray Research, Inc., Mendota Heights, MN, and with the Cray Y-MP computer operated by the Center for High Performance Computing of The University of Texas System. The cooperation of each of these centers in making their facilities available is greatly appreciated.

The authors wish to acknowledge the Center for Numerical Analysis and the Computation Center of The University of Texas at Austin for their support. In addition, this work was supported, in part, by the National Science Foundation under grant C.R.-8515722, the Department of Energy under grant DE-FG05-88ER25048, and Cray Research, Inc., under grant LTR DTD with The University of Texas at Austin.

REFERENCES

Rectilinear Partitioning of Irregular Data Parallel Computations

David M. Nicol
Department of Computer Science
College of William and Mary
Williamsburg, VA 23185 USA
Internet: nicol@cs.wm.edu

Introduction

This paper describes new mapping algorithms for domain-oriented data-parallel computations, where the workload is distributed irregularly throughout the domain. We consider the problem of partitioning the domain (represented as an \( n \times m \) array of execution weights) for an \( N \times M \) array of locally connected parallel processors, in such a way that the workload on the most heavily loaded processor is minimized, subject to the constraint that the partition be perfectly rectilinear, as illustrated by Figure 1. The total execution weight of a processor is the sum of all execution weights assigned to its rectangle. Data parallel computations tend to have localized data dependencies; rectilinear partitions ensure that all communication induced by localized data dependencies is between local processors; this algorithm will be particularly useful on architectures where there is a high difference between the cost of local and global communication. We will not explicitly include communication costs in our model. Our implicit assumption is that communication costs will be minimized when locality is ensured.

This paper provides an improved algorithm for finding the optimal partition in one dimension; new algorithms for partitioning in two dimensions, and shows that optimal partitioning in three dimensions is NP-complete. We discuss our application of these algorithms to real problems.

Figure 1. Rectilinear Partitioning of Two Dimensional Domain

One Dimensional Partitioning

The rectilinear partitioning algorithm in one dimension has been extensively studied as the chains-on-chains partitioning problem [1, 3, 4, 5]; we are given a linear sequence of work pieces (called modules), and wish to partition the sequence for execution on a linear array of processors. The best known published algorithm to date finds the optimal partitioning in \( O(Mm \log m) \) time, where \( M \) is the number of processors and \( m \) is the number of modules. This solution repeatedly calls a probe function. This function accepts an argument \( W \), and uses a greedy workload assignment algorithm to determine whether there is a partition that assigns no more than \( W \) work to each processor. A probe is used in conjunction with a search, in order to find the minimal \( W \) for which there exists a feasible partition. The cost of calling probe is \( O(Mm \log m) \). Previous solutions have involved calling probe \( O(m) \) times. We have a new search strategy that reduces the calling frequency to \( O(Mm \log m) \), thereby reducing the complexity of one-dimensional partitioning to \( O(m^2 + Mm^2 \log^2 m) \).

Two Dimensional Partitioning

The heart of our 2D partitioning algorithm is an ability to optimally partition in one dimension, given a fixed partition in the other. Suppose a row partition \( R \) is given. We can compress workload forced by \( R \) to reside on a common processor into super-pieces, thereby creating an \( N \times M \) load matrix. This matrix can be viewed as \( N \) one dimensional chains, a common partitioning of their columns will produce a 2D rectilinear partition.

The problem of finding an optimal column partition can be approached through a minor modification to the 1D probe function. This modification raises the cost of calling probe to \( O(Mm \log m) \), but is a cheaper preprocessing step. Otherwise, the conditionally optimal problem is solved in the same way as the 1D partitioning problem, in \( O(mn + Mm^2 \log^2 m) \) time.

We may apply the conditionally optimal partitioning algorithm in an iterative fashion. Suppose that a row partition \( R \) is given. For example, we might construct an initial row partition as follows. Sum the weights of all work pieces in a common row, to create a super-piece representing that row. Find an optimal 1D partition of these super-pieces onto \( N \) processors. Use this partition as \( R_b \) assume \( R_a \) is to be fixed, and let \( C \) be the optimal column partition, given \( R \). Let \( z = \pi(R_a, C) \) be the cost of that partitioning. Next, fix the column partition as \( C_i \), and let \( R \) be the optimal row partitioning, given \( C_i \). Clearly we may repeat this process as many times as we like. We have shown that the sequence \( z_1, z_2, \ldots, z \) is monotone non-increasing, and that eventually the computation converges to a fixed row partition \( R \) and column partition \( C \). We have also bounded the number of iterations required for convergence by \( O(Mm^2 + n + m) \). Far fewer iterations are required to converge, in practice. The talk will discuss the use of this procedure on highly irregular 2D grids used in fluid-flow problems.

Three Dimensional Partitioning

Finally, we consider the complexity of the 3D partitioning problem. We have already seen that the 1D problem can be solved in polynomial time; it is not yet known whether the 2D problem is tractable. It turns out that the problem of finding an optimal rectilinear partition in three dimensions is NP-complete. The proof shows that the monotone 3SAT problem [6] can be reduced to rectilinear partitioning in three dimensions. The key idea is to construct a domain as a function of the 3SAT clauses. Each literal is given three rows, or columns in the 3D weight matrix. The intersection of rows and columns for literals \( x_1, x_2, x_3 \) is a \( 3 \times 3 \times 3 \) volume. This volume is to be partitioned among \( 2 \times 2 \times 2 \) processors, which essentially forces each literal group to divide into one of two possible partitionings. The partitioning choice can be interpreted as the assignment of a truth value to the literal. The volume is weighted in such a way that its partition has bottleneck value 1 if and only if the partition corresponds to an assignment that satisfies the clause. This shows that optimal three dimensional partitioning is as hard as the monotone 3SAT problem, which is known to be NP-complete.

References


---

This research was supported in part by the Army Aeronautics Research and Development Activity through NASA grant NAG-1-777, in part by NASA grant NGL-1-1132, and in part by NSF Grant ASC 8415373.
Applying Chain Mapping Algorithms to Flowgraphs

David R. O’Hallaron

School of Computer Science
Carnegie Mellon University, Pittsburgh, PA, 15213, USA

Abstract

We identify two useful classes of flowgraphs that can be compiled onto distributed-memory multicomputers using chain mapping algorithms, and whose performance can be predicted accurately.

1 Introduction

Flowgraphs have been used for years to model digital signal processing (DSP) algorithms. A flowgraph is a collection of nodes and arcs, where nodes represent computations and arcs represent FIFO queues. Each node iterates an infinite number of times; each iteration, a node consumes a vector of data items from each of its input arcs, performs a computation on the input vectors, and produces a vector of data items on each of its output arcs. The sizes of the input and output vectors are indicated by integer arc labels.

There is much published research on the problems of compiling flowgraphs onto sequential machines and onto shared-memory multiprocessors[3]. Little attention has been given to date on the problem of compiling flowgraphs onto distributed-memory multicomputers; some recent work can be found in [5, 8].

In this paper we identify two useful classes of flowgraphs that we call pipes and trellises. A pipe models a pipelined sequence of different operations performed on a single data stream. A trellis models a single operation performed in parallel on different data streams. Pipes and trellises are interesting because they can be compiled efficiently and optimally using chain mapping algorithms, and because their performance can be accurately predicted.

2 Pipes and Trellises

A common DSP application is to pipeline data from a sensor through a sequence of filtering operations such as FIR filters and FFT’s. Such applications are modeled by a flowgraph we call a pipe. An example with 4 filtering operations is shown in Figure 1.

In another common DSP application, data from multiple sensors is distributed, processed independently and combined in some way.

Such applications can be modeled by a flowgraph we call a trellis. An example with four sensors is shown in Figure 2.

Figure 2: A trellis.

Pipes and trellises are extremely useful classes of flowgraphs that can be used to build real applications. For example, sonar adaptive beam interpolation and the 2D FFT can each be modeled as a pipe of two trellises.

3 Applying Chain Methods

Chain mapping algorithms[1, 2, 4] compile a chain of M modules onto a chain of P distributed-memory processors, typically with P ≤ M. These algorithms are attractive because they are efficient, requiring time polynomial in M and P, and because they are optimal, minimizing the maximum load on any processor, subject to the constraint that two contiguous modules are mapped on the same processor or its neighbor.

Pipes and trellises have the nice property that they are easily transformed into a form suitable for chain mapping methods. The transformation of pipes is trivial; each node in the flowgraph becomes a module for the chain mapping algorithm. This is shown in Figure 3. The transformation for trellises is also straightforward, as shown in Figure 4. Each of the dashed boxes becomes a module for the chain mapping algorithm.

Figure 3: Transformation of a pipe, M = 4.
4 Predicting Performance

We have seen that pipes and trellises can be compiled onto distributed-memory multicomputers using chain mapping methods. Another nice quality of these flowgraphs is that their performance can be predicted quite accurately. A common performance measure for flowgraphs is speedup, denoted by \( S \) and defined by \( S = E/P \), where \( E \) denotes efficiency. We derive an expression for the efficiency of pipes and trellises under the following assumptions: (1) All modules consume the same number of clocks per iteration. (2) The number of modules \( M \) is an integral multiple of the number of processors \( P \).

Let \( T = T_c + T_d \) be the number of clocks required to transfer data between modules, where \( T_c \) is the number of clocks spent doing computations and \( T_d \) is any additional clocks required because of interprocessor communication, these can be clocks spent waiting for data or actually transferring data between a communications network and memory. Under these assumptions, efficiency is simply

\[
E = \frac{T_c}{T_c + T_d} = \frac{1}{1 + T_d/T_c}
\]

(1)

If the computations associated with each node in the flowgraph are data-independent, true for most DSP operations, then \( T_c \) can be predicted precisely for a given parallel computer and compiler. On the other hand, \( T_d \) may be more difficult to predict accurately because it could include time spent waiting for data to arrive. For our model, we will assume that \( T_d = 2N \), where \( N \) is the number of inputs (and outputs) for each module, and \( T_c \) is the overhead (in clocks) associated with transferring a word between the communication network and memory. And finally, notice also that this expression for \( T_d \) is a lower bound that ignores any clocks spent waiting for data to arrive. While this is unrealistic in general, empirical evidence suggests that this is reasonable for pipes and trellises. Given the expression for \( T_c \), efficiency becomes

\[
E = \frac{1}{1 + 2N/T_c}
\]

(2)

We tested the model in (2) using four different pipes and trellises running on a 64-processor iWarp computer at Carnegie Mellon[6, 7]. Each flowgraph was compiled using a chain mapping algorithm, and the resulting chain was embedded in the iWarp's 2D mesh. The results are shown in Figure 5. Predicted efficiency was computed using (2). For all tests, \( P = 64 \), \( M = 64 \), and \( r = 40 \). Measured efficiency was obtained by measuring the speedup \( S \) and then applying the identity \( E = S/P \).

5 Discussion

The empirical results are somewhat startling. In each case, the predicted performance was within one percent of the measured performance. Communications overhead was restricted to the unavoidable cost of physically transferring data between the communications network and memory. There was negligible overhead due to waiting for data.

The accuracy of the model allows us to make strong predictions about performance for various values of \( r \), \( N \), and \( T_c \). For example, if \( r \) could be reduced from 40 to 4, which is quite possible for iWarp, then the efficiency of the trellis flowgraph in Figure 5 would rise from 34% to 84%.

References


<table>
<thead>
<tr>
<th>Graph</th>
<th>( N )</th>
<th>( T_c )</th>
<th>Predicted ( E )</th>
<th>Measured ( E )</th>
</tr>
</thead>
<tbody>
<tr>
<td>pipe1</td>
<td>1024</td>
<td>55N</td>
<td>41%</td>
<td>40%</td>
</tr>
<tr>
<td>pipe2</td>
<td>1024</td>
<td>23N^2</td>
<td>99%</td>
<td>99%</td>
</tr>
<tr>
<td>trellis1</td>
<td>2048</td>
<td>42N</td>
<td>34%</td>
<td>34%</td>
</tr>
<tr>
<td>trellis2</td>
<td>2048</td>
<td>41N + 562(N/K)^2</td>
<td>80%</td>
<td>80%</td>
</tr>
</tbody>
</table>

Figure 5: Predicted and measured efficiency on iWarp
A MAPPING ALGORITHM FOR HETEROGENEOUS MULTIPROCESSOR ARCHITECTURES

Todd P. Carpenter
Honeywell Systems and Research Center
3660 Technology Drive
Mpls, MN 55418-1093 USA
carpent@src.honeywell.com

Sudhakar Yalamanchili
School of Electrical Engineering
Georgia Institute of Technology
Atlanta, GA. 30332-0250, USA
sudha@ececm.gatech.edu

Abstract-Resource management is an integral aspect of the design, development, and application of multiprocessor architectures. This paper is concerned with one specific aspect of the problem - i.e., assignment of concurrently executing tasks to the processors. Naive assignments can lead to excessive performance degradation due to the consequent overhead in inter-task communication and synchronization. In particular, we focus on heterogeneous architectures comprised of distinct processor types, distinct communication media, and many data types. Such systems are becoming increasingly wide-spread, particularly in special purpose applications and distributed systems. We address the problem of statically mapping programs to such multiprocessors.

1. INTRODUCTION

Examples of heterogeneous applications include distributed applications - different processor platforms, mix of workstations, minis, minisuper, microframe computers, etc. They represent different processors of varying performance and compilers may or may not exist for all languages on all machines. Further, each application module runs at differing speeds on different processors. Each pair of processors may communicate at different rates. For instance, heterogeneous environments might consist of Apollo/SUN, Datastations, Vaxen and Amiga platforms. Alternatively, dedicated applications often are supported by heterogeneous multiprocessor architectures to realize high performance. For example, The Intel IPSC/2 hypercube is available as mixed-speed systems using 1860 and 80386 processors. Packaging constraints often lead to systems where subcubes of a hypercube residing in different racks communicate over different media (e.g., fiber) to processors within the rack. Special purpose multiprocessor architectures for signal processing have been designed with hierarchically organized interconnection networks with differing cost/delay characteristics at each level.

The problem of mapping parallel programs onto homogeneous architectures is a sufficiently complex combinatorial problem in its own right [1]. The presence of additional constraints on processor and processing types and variable communication delays makes the problem even more difficult. In this paper we describe an approach for realizing such assignments. This approach is an adaptation of an approach we successfully implemented for computing assignments when the target architecture (and the computations) are homogeneous [2]. This work only applies to the static assignment problem. We do not yet address the more difficult dynamic mapping problem.

II. MODEL AND METHODS

The mapping problem is addressed in the context of exploiting medium-coarse grained parallelism in message passing, multiple instruction streams, multiple data streams (MIMD) architectures. The application and architecture are described in terms of attributed directed graphs. Node (edge) weights represent communication (computation) requirements. Nodes are also labeled by a type representing the nature of the computation, e.g., floating point, symbolic, etc. Assignments must now maintain compatibility between types, i.e., a node of a certain type is constrained to execute on only a subset of processors. We propose to compute assignments based on an adaptation of simulated annealing, a combinatorial optimization procedure that realizes a probabilistic search of a discrete state space [3]. The process starts from some initial state which is perturbed. The new state is evaluated based on an objective function (also known as the energy function). If this state is "better" as measured by the value of the energy function it is accepted; if not it is probabilistically accepted. It is this latter, random behavior that enables the search process to avoid being trapped in local minima. Simulated annealing has found widespread application in solving combinatorial optimization problems. A simplified view of the process is shown below, and each step is elaborated in the following.

Loop until done
State perturbation;
State evaluation;
State acceptance/rejection;
End loop

A. State Perturbation

Given an assignment of tasks to processors, this step modifies the assignment(s) of task(s). The change may be based on architectural characteristics (e.g., limited to adjacent processors) or may be based on the structure of algorithm (e.g., limited to communicating tasks). In a single iteration, multiple tasks may be moved simultaneously corresponding to large jumps in the state space, or one task may be moved at a time. Further, the perturbation strategies may change over time to adapt to the behavior of the search. Finally, the perturbations are governed by type constraints, i.e., between tasks and processors.

B. State Evaluation

The energy function maps states to real integer values. The value of the function is a measure of the quality of the mapping represented by the state. For the mapping problems we are considering examples of appropriate energy functions include maximum-processor load, maximum-interprocessor communication load, variance of processor loads, etc. The more complex the energy function the longer the run-times, or the smaller number of states evaluated per unit time. The goal is to find a state that maximizes (or minimizes) the value of the energy function.

We propose an energy function that we have successfully used when mapping onto homogeneous architectures. This function consists of two components - a load balancing component and a inter-processor communication component. It is typically desirable to maximize the first and minimize the second, and can be represented as 

\[ E = w_1 \times LB + w_2 \times C \]

where \( w_1, w_2 \) are the weighting factors for load balancing and inter-processor communications. The individual components are computed as follows.
Load balance is a measure of how evenly the processing load is distributed among the processors. The objective is to distribute the processing load as evenly as possible. The value of the expression $LoadBalance$ shown below is minimized when the load is evenly distributed.

Define:

\[ P = \text{number of resource elements} \]
\[ T_i = \text{number of tasks mapped to element } i \]
\[ Tw_{ij} = \text{the weight of the } j \text{th task mapped to resource element } i \]
\[ Pwi_{i,\text{type}} = \text{the weight for resource element to perform type task} \]
\[ T_{ij} = \text{the type of task } j \text{ mapped to element } i \]
\[ LR = \text{the number of resource elements with nonzero loads} \]

\[ LoadBalance = \frac{1}{LR} \sqrt{\sum_{i=1}^{P} \left( \sum_{j=1}^{T_i} (Tw_{ij} \times Pwi_{i,\text{type}}) \right)} \]

The expression $Commisions$ shown below is a measure of how the interprocessor communications load is distributed. The load is a function of the distance between communicating tasks, the amount of data, and the link throughput. The value of $Commisions$ is minimized when all tasks are mapped to the same processor. Therefore, this expression is at odds with the load balancing term described above.

Define:

\[ T = \text{number of tasks} \]
\[ T_{Oj} = \text{number of outputs of task } T_i \]
\[ L_{Rxi}L_{Rrj} = \text{the links from resource to which Task } i \text{ is mapped, to the resource for task } j \]
\[ T_{ci,j} = \text{the amount of data from Task } i \text{ to Task } j \]
\[ L_k = \text{reciprocal of throughput of link } k \]
\[ \beta = \text{Amount of communications bandwidth width} \]

\[ Commisions = \frac{\sum_{i=1}^{T} \sum_{j=1}^{T_{Oj}} \sum_{L_{Rxi}L_{Rrj}} T_{ci,j} \times L_k}{\beta} \]

C. State Acceptance/Rejection

New states are accepted or rejected based on the value of the energy function. We deviate from traditional annealing techniques in the decision process. The following acceptance/rejection techniques were employed in our study. Note that although we can accept worse states, we do maintain a record of the minimum state found.

- Annealing: If $AE \leq 0$, the new state is accepted, and is used as the starting point for the next iteration. If $AE > 0$, the new state is accepted probabilistically according to $P(AE) = e^{-\frac{AE}{k}}$ where $k$ is the Boltzmann's constant and $T$ is the temperature.

- Kappa Sequence: If $AE \leq 0$, or if $\kappa$ contiguous states have been rejected, accept the new state. The value of $\kappa$ is increased if $AE < 0$, and is decreased if research will expand the role of the mapping algorithm to consider its effect on reliability, and the feasibility of computing assignments that optimize other performance attributes such as response time.

V. FUTURE WORK

This work is part of a larger effort to develop a multiprocessor system toolkit to support the conception, design, analysis, and development of large scale, medium-to-coarse grained multiprocessor architectures. Future work will expand the role of the mapping algorithm to consider its effect on reliability, and the feasibility of computing assignments that optimize other performance attributes such as response time.

Experience in Using SIMD and MIMD Parallelism for Computational Fluid Dynamics

Horst D. Simon*  
Applied Research Branch  
NASA Ames Research Center, Mail Stop T045-1  
Moffett Field, CA 94035

Leonardo Dagum*  
Applied Research Branch  
NASA Ames Research Center, Mail Stop T045-1  
Moffett Field, CA 94035

March 20, 1991

Abstract One of the key objectives of the Applied Research Branch in the Numerical Aerodynamic Simulation (NAS) Systems Division at NASA Ames Research Center is the accelerated introduction of highly parallel machines into a full operational environment. In this report we summarize some of the experiences with the parallel testbed machines at the NAS Applied Research Branch. We discuss the performance results obtained from the implementation of two Computational Fluid Dynamics (CFD) applications, an unstructured grid solver and a particle simulation, on the Connection Machine CM-2 and the Intel iPSC/660.

Keywords: parallel architectures, parallel algorithms
AMS Subject Classification: J.6.08, 65W05, 65N39.
CR Subject Classification: G.1.8, J.2, C.1.1, C.1.2.

1 Introduction

One of the key tasks of the Applied Research Branch in the Numerical Aerodynamic Simulation (NAS) Systems Division at NASA Ames Research Center is the accelerated introduction of highly parallel and related key hardware and software technologies into a full operational environment (see [1]). From 1988 to 1990 a testbed facility has been established for the development and demonstration of highly parallel computer technologies. Currently a 32k processor Connection Machine CM-2 and an 128 node Intel iPSC/660 are operated at the NASA Applied Research Branch. This testbed facility is envisioned to consist of successive generations of increasingly powerful highly parallel systems that are scalable to high performance capabilities beyond that of conventional supercomputers.

It is recognized within the scientific computing community that the most promising approach toward achieving very large improvements in computing performance is through the application of highly parallel architectures. To meet the future processing needs of the aerospace research community, the Applied Research Branch supports a research program aimed at achieving the best match of parallel processing technology to the most demanding research applications. In the last two years a number of large scale computational fluid dynamics applications have been implemented on the two testbed machines, and the potential of the parallel machines for production use has been evaluated. Beyond that, a systematic performance evaluation effort has been initiated (see [4]), and basic algorithm research has been continued.

In this report we will first give a brief description of the capabilities of the parallel machines at NASA Ames. Then we will discuss some of the research carried out in the implementation of Computational Fluid Dynamics (CFD) applications on these parallel machines. We focus here on those applications where we have more detailed knowledge because of our own involvement, an explicit 2D Euler solver for unstructured grids, and a simulation based on particle methods. Other applications based on structured grids will be mentioned briefly, as well as the NAS effort in parallel benchmarking. In a final section we offer some preliminary conclusions on the performance of commercial parallel machines for CFD applications, as well as the potential of the different architectures for production use in the future. Another summary of some of the results from NASA Ames is given by D. Bailey in [3].

2 Parallel Machines at NASA Ames

2.1 Connection Machine

The Thinking Machines Connection Machine (CM) is a massively parallel SIMD computer consisting of thousands of bit serial data processors under the direction of a front end computer. The system at NASA Ames consists of 32768 bit serial processors each with 1 Mbit of memory and operating at 7 MHz. The processors and memory are packaged as 16 in a chip. Each chip also contains the routing circuitry which allows any processor to send and receive messages from any other processor in the system. In addition, there are 1024 64-bit Weitek floating point processors which are fed from the bit serial processors through a special purpose "Sprint" chip. There is one Sprint chip connecting every two CM chips to a Weitek. Each Weitek processor can execute an add and a multiply each clock cycle thus performing at 14 MFLOPS and yielding a peak aggregate performance of 14 GFLOPS for the system.

The Connection Machine can be viewed two ways, either as an 11-dimensional hypercube connecting the 2048 CM chips or a 10-dimensional hypercube connecting the 1024 processing elements. The first view is the "fieldwire" model of the machine which has existed since its introduction. This view admits to the existence of at least 32768 physical processors (when using the whole machine) each storing data in fields within its local memory. The second is the more recent "sliceview" model of the machine which admits to only 1024 processing elements (when using the whole machine) each storing data in slices of 32 bits distributed across the 32 physical processors in the processing element. Both models allow for "virtual processing", where the resources of a single processor or processing element may be divided to allow a greater number of virtual processors.

Regardless of the model, the architecture allows interprocessor communication to proceed in three manners. For very general communication with no regular pattern, the router determines the destination of messages at run time and directs the messages accordingly. This is referred to as general router communication. For communication with an irregular but static pattern, the message paths may be pre-computed and the router will direct messages according to the pre-computed paths. This is referred to as compiled communication and can be 5 times faster than general router communication. Finally, for communication which is perfectly regular and involves only shifts along grid axes, the system software optimizes the data layout by ensuring strictly nearest neighbor communication and uses its own pre-computed paths. This is referred to as NEWS (for "NorthEastWestSouth") communication. Despite the name, NEWS communication is not restricted to 2 dimensional grids, and up to 3 dimensional NEWS grids may be specified. NEWS communication is the fastest.

The I/O subsystems connect to the data processors through an I/O controller. An I/O controller connects to 8, 32 processors through 256 I/O lines. There is one line for each chip but the controller can only connect to 256 lines simultaneously and must treat its 8 processors as two banks of 4k each. Each I/O controller allows transfer rates of up to 40 MB per second. In addition to an I/O controller there can be a frame buffer for video graphics output. Because it is connected directly to the backplane rather than through the I/O bus, the frame buffer can receive data from the CM processors at 256 MB per second. The
We discuss here work on an upwind finite-volume flow solver for the Eu-

2.2 Intel iPSC/860

The Intel iPSC/860 (also known as Touchstone Gamma System) is

3 Structured Grid Applications

Structured grid codes, in particular multiblock structured grid codes,

4 Unstructured Grid Applications

We discuss here work on an upwind finite-volume flow solver for the Eu-

References
4.1 SIMD Implementation of Unstructured Solver

For the implementation on the CM-2, Hammond and Barth [11] used a novel partitioning of the problem which minimizes the computation and communication costs on a massively parallel computer. In a mesh vertex scheme, solution variables are associated with each vertex of the mesh and flux computation is performed at edges of the non-overlapping control volumes which surround each vertex. In conventional parallel implementations this operation is partitioned to be performed edge-wise, i.e., each edge of the control volume is assigned to one processor (edge-based). The resulting flux calculation contributes to two control volumes which share the particular edge.

In the partitioning used by Hammond and Barth, each vertex of the mesh is assigned to one processor (vertex-based). Flux computations are identical to the edge-based scheme but computed by processors associated with vertices. Each edge of the mesh joins a pair of vertices and is associated with one edge of the control volume.

One can direct edge \((i,j)\) to determine which vertex in the pair computes the flux through the shared edge of the control volume, \((k',j')\). When there is a directed edge from \(i\) to \(j\), then the processor holding vertex \(j\) sends its conserved values to the processor holding vertex \(i\), and the flux across the common control volume edge is computed by processor \(i\) and accumulated locally. The flux through \((k',j')\) computed by the processor holding vertex \(i\) is sent to the processor holding vertex \(j\) to be accumulated negatively. Hammond and Barth show that their vertex-based scheme requires 50% less communication and asymptotically identical amounts of computation as compared with the traditional edge-based approach.

Another important feature of the work by Hammond and Barth is the use of fast communication. A feature of the communication within the flow solver here is that the communication pattern, although irregular, remains static throughout the duration of the computation. The SIMD implementation takes advantage of this by using a mapping technique developed by Hammond and Schreiber [12] and a "Communication Compiler" developed for the CM-2 by Dahl [10]. The former is a highly parallel graph mapping algorithm that assigns vertices of the grid to processors in the computer such that the sum of the distances that messages travel is minimized. The latter is a software facility for scheduling irregular communications with a static pattern. The use specifies a list of source locations and destinations for messages which are then compiled into routing paths to be used at run-time.

Hammond and Barth have incorporated the mapping algorithm and the communication compiler into the flow solver running on the CM-2 and have realized a factor of 30 reduction in communication time compared to using naive or random assignments of vertices to processors and the router. Using 8K processors of the CM-2 and a VP ratio of 2, Hammond and Barth carried out 109 time steps of the flow solver in about 71.62 seconds. This does not include setup time.

4.2 MIMD Implementation of Unstructured Solver

Similar to the SIMD implementation one of the key issues is the partitioning of the unstructured mesh. In order to partition the mesh Venkatkrishnan et al. [23] employ a new algorithm for the graph partitioning problem, which has been discussed recently by Simon [22], and which is based on the computation of eigenvectors of the Laplacian matrix of a graph associated with the mesh. Details on the theoretical foundations of this strategy can be found in [19]. Detailed investigations and comparisons to other strategies (cf. [22]) have shown that the spectral partitioning produces subdomains with the shortest boundary, and hence tends to minimize communication cost.

After the application of the partition algorithm of the previous section, the whole finite volume grid with triangular cells is partitioned into \(P\) subgrids, each subgrid contains a number of triangular cells which form a single connected region. Each subgrid is assigned to one processor. All connectivity information is precomputed, using sparse matrix type data structures.

Neighboring subgrids communicate to each other only through their interior boundary vertices which are shared by the processors containing the neighboring subgrids. In the serial version of the scheme, field quantities (mass, momentum and energy) are initialized and updated at each vertex of the triangular grid using the conservation law for the Euler equations applied to the dual cells. Each processor performs the same calculations on each subgrid as it would do on the whole grid in the case of a serial computation. The difference is that now each subgrid may contain both physical boundary edges and interior boundary edges, which have resulted from grid partitioning. Since a finite volume approach is adopted, the communication at the interprocessor boundaries consists of summing the local contributions to integrals such as volumes, fluxes, gradients etc.

The performance of the Intel iPSC/860 on the test problem is given in Table 1.

| Table 1: Performance of Unstructured Grid Code on the Intel iPSC/860 |
|-----------------|---------|--------|--------|
| Processors     | secs/step | MFLOPS | efficiency(%) |
| 2              | 7.58     | 7.7    | 83     |
| 4              | 3.82     | 15.3   | 83     |
| 8              | 2.01     | 29.1   | 79     |
| 16             | 1.11     | 52.7   | 71     |
| 32             | 0.61     | 95.9   | 65     |
| 64             | 0.33     | 177.3  | 60     |
| 128            | 0.21     | 270.5  | 47     |

5 Particle Methods

Particle methods of simulation are of interest primarily for high altitude, low density flows. When a gas becomes sufficiently rarefied the constitutive relations of the Navier-Stokes equations (i.e. the Stokes law for viscosity and the Fourier law for heat conduction) no longer apply and either higher order relations must be employed or the continuum approach must be abandoned and the molecular nature of the gas must be addressed explicitly. The latter approach leads to direct particle simulation.

In direct particle simulation, a gas is described by a collection of simulated molecules thus completely avoiding any need for differential equations explicitly describing the flow. By accurately modelling the microscopic State of the gas the macroscopic description is obtained through the appropriate integration. The primary disadvantage of this approach is that the computational cost is relatively large. Therefore, although the molecular description of a gas is accurate at all densities, a direct particle simulation is competitive only for low densities where accurate continuum descriptions are difficult to make.

For a small discrete time step, the molecular motion and collision terms of the Boltzmann equation may be decoupled. This allows the simulated particle flow to be considered in terms of two consecutive but distinct events in one time step, specifically there is a collisionless motion of all particles followed by a motionless collision of those pairs of particles which have been identified as colliding partners. The collisionless motion of particles is strictly deterministic and reversible. However, the collision of particles is treated on a probabilistic basis. The particles move through a grid of cells which serves to define the geometry, to identify colliding partners, and to sample the macroscopic quantities used to generate a solution.

The state of the system is updated on a per time step basis. A single time step is comprised of five events.

1. Collisionless motion of particles.
2. Enforcement of boundary conditions.
3. Pairing of collision partners.
4. Collision of selected collision partners.
5. Sampling for macroscopic flow quantities.

Detailed description of these algorithms may be found in [17] and [18].
5.1 SIMD Implementation of Particle Simulation

Particle simulation is distinct from other CFD applications in that there are two levels of parallel granularity in the model. There is a coarse level consisting of cells in the simulation (which are approximately equivalent to grid points in a continuum approach) and there is a fine level consisting of individual particles. At the time of the CM-2 implementation there existed only the fieldwise model of the machine, and it was natural for Dagum [7] to decompose the problem at the finest level of granularity. In this decomposition, the data for each particle is stored in an individual virtual processor in the machine. A separate set of virtual processors (or VP set) stores the geometry and yet another set of virtual processors stores the sampled macroscopic quantities.

This decomposition is conceptually pleasing however in practice the relative slowness of the Connection Machine router can prove to be a bottleneck in the application. Dagum [7] introduces several novel algorithms to minimize the amount of communication and improve the overall performance in such a decomposition. In particular, steps 2 and 3 of the particle simulation algorithm require a somewhat less than straightforward approach.

The enforcement of boundary conditions requires particles which are about to interact with a boundary to get the appropriate boundary information from the VP set storing the geometry data. Since the number of particles undergoing boundary interaction is relatively small, a master/slave algorithm is used to minimize both communication and computation. In this algorithm, the master is the VP set storing the particle data. The master creates a slave VP set large enough to accommodate all the particles which must undergo boundary interactions. Since the slave is much smaller than the master, instructions on the slave VP set execute much faster. This more than makes up for the time that the slave requires to get the geometry information and to both get and return the particle information.

The pairing of collision partners requires sorting the particle data such that particles occupying the same cell are represented by neighboring virtual processors in the one dimensional NEWS grid storing this data. Dagum [8] describes different sorting algorithms suitable for this purpose. The fastest of these makes use of the realization that the particle data moves through the CM processors in a manner analogous to the motion of the particles in the simulation. The mechanism for disorder is the motion of particles, and the extent of motion of particles, over a single time step, is small. This can be used to tremendously reduce the amount of communication necessary to re-order the particles.

These algorithms have been implemented in a two-dimensional particle simulation running on the CM-2. At the time of implementation, the CM-2 at NASA Ames had only 64k bits of memory per processor which was insufficient to warrant a three-dimensional implementation. Therefore, the slicewise model of the machine did not exist and the machine had the slower 32-bit Westek's which did not carry out any integer arithmetic. Nonetheless, with this smaller amount of memory and fieldwise implementation, the code was capable of simulating over $2.0 \times 10^6$ particles in a grid with $6.0 \times 10^3$ at a rate of 2.0usec/particle/timestep using all 32k processors (see [7]). By comparison, a fully vectorized equivalent simulation on a single processor of the Cray YMP runs at 1.0usec/particle/timestep and 86 MFLOPS as measured by the Cray hardware performance monitor. (Note that a significant fraction of a particle simulation involves integer arithmetic and the MFLOPS measure is not completely indicative of the amount of computation involved.) Currently, work is being carried out to extend the simulation to three dimensions using a parallel decomposition which takes full advantage of the slicewise model of the machine.

Table 2: Performance of Particle Simulation on the Intel iPSC/860

<table>
<thead>
<tr>
<th>Processors</th>
<th>µsec/prt/step</th>
<th>MFLOPS</th>
<th>efficiency (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>24.4</td>
<td>3.5</td>
<td>97</td>
</tr>
<tr>
<td>4</td>
<td>12.5</td>
<td>6.9</td>
<td>95</td>
</tr>
<tr>
<td>8</td>
<td>6.35</td>
<td>13.5</td>
<td>93</td>
</tr>
<tr>
<td>16</td>
<td>3.25</td>
<td>26.5</td>
<td>91</td>
</tr>
<tr>
<td>32</td>
<td>1.63</td>
<td>52.8</td>
<td>91</td>
</tr>
<tr>
<td>64</td>
<td>0.85</td>
<td>101</td>
<td>87</td>
</tr>
<tr>
<td>128</td>
<td>0.42</td>
<td>215</td>
<td>88</td>
</tr>
</tbody>
</table>

In McDonald's [18] implementation, the spatial domain of the simulation is divided into a number of sub-domains or regions equal to the desired number of node processes. Communication between processes occurs as a particle passes from one region to another and is carried out asynchronously, thus allowing overlapping communication and computation. Particles crossing region "seams" are treated simply as an additional type of boundary condition. Each simulated region of space is surrounded by a shell of extra cells that, when entered by a particle, directs that particle to the neighboring region. This allows the representation of simulated space (i.e. the geometry definition) to be distributed along with the particles. The aim is to avoid maintaining a representation of all simulated space which, if stored on a single processor, would quickly become a serious bottleneck for large simulations, and if replicated would simply be too wasteful of memory.

Within each region the sequential or vectorized particle simulation is run. This applies to decomposing for great flexibility in the physical model that is implemented since node processes are asynchronous and largely independent of each other. Recall that communication between processes is required only when particles cross region seams. This is very fortuitous since the particle motion is straightforward and fully agreed upon. The important area of research has to do with the modelling of particles, and since this part of the problem does not directly affect communication, particle models can evolve without requiring great algorithmic changes.

McDonald's implementation is fully three-dimensional. The performance of the code on a 3D heat bath is given in Table 2.

At the present time the domain decomposition is static, however work is being carried out to allow dynamic domain decomposition thus permitting a good load balance to exist throughout a calculation. The geometry and spatial decomposition of the heat bath simulation generated the area to volume ratio of the regions in order to more closely approximate the communication expected in a real application with dynamic load balancing. The most promising feature of these results is the linear speed up obtained, indicating that the performance of the code should continue to increase with increasing numbers of processors.

6 Conclusions

On the unstructured grid code the performance figures are summarized in Table 3, where all MFLOPS numbers are Cray Y-MP equivalent numbers.

Table 3: Performance Comparison of Unstructured Grid Code

<table>
<thead>
<tr>
<th>Machine</th>
<th>Processors</th>
<th>secs/step</th>
<th>MFLOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cray Y-MP</td>
<td>1</td>
<td>0.30</td>
<td>150.0</td>
</tr>
<tr>
<td>Intel iPSC/860</td>
<td>64</td>
<td>0.33</td>
<td>177.3</td>
</tr>
<tr>
<td>CM-2 (32 bit)</td>
<td>128</td>
<td>0.21</td>
<td>278.6</td>
</tr>
</tbody>
</table>

For the particle methods the corresponding summary of performance figures can be found in Table 4. The figures in Table 4 should be interpreted very carefully. The simulations run on the different machines were comparable, but not identical. The MFLOPS are Cray Y-MP...
The results in Tables 3 and 4 demonstrate a number of points. Both unstructured grid computations and the particle simulations are applications which a priori are not immediately parallelized, and for which both on SIMD and MIMD machines considerable effort must be expended in order to obtain an efficient implementation. It has been demonstrated by the results obtained at NASA Ames that this can be done, and that supercomputer level performance can be obtained on current generation parallel machines. Furthermore the particle simulation code on the CM-2 is a production code currently used to obtain production results (see [9]). The iPSC/860 implementation should be in production use by the end of 1991.

Our results also demonstrate another feature which has been found across a number of applications at NASA Ames: massively parallel machines quite often obtain only a fraction of their peak performance on realistic applications. In the applications considered here, the requirement for unstructured, general communication has been the primary impediment in obtaining the peak realizable performance from these machines. Neither the CM-2 nor the the iPSC/860 deliver the communication bandwidth necessary for these CFD applications. This situation is even worse for implicit algorithms (see e.g. [2]). Experience has shown that CFD applications require on the order of one memory reference per floating point operation and a balanced system should have a memory bandwidth comparable to its floating point performance. In these terms, current parallel systems deliver only a fraction of the required bandwidth.

Acknowledgement. We wish to thank our colleagues with the NAS Applied Research Branch, whose work has been discussed here: D. Bailey, E. Barszcz, R. Fatoohi, T. Lasinski, C. Levit, V. Venkatakrishnan, and S. Weeratunga. We also thank T. Barth, D. Jespersen, J. McDonald, (all NASA Ames) and P. Frederiksen, S. Hammond, and R. Schreiber at RIACS for their contributions to this summary report.

References


AUTOMATIC PARTITIONING OF
FINITE-ELEMENT/FINITE-DIFFERENCE
MESHES FOR PARALLEL PROCESSING

Charbel Farhat
Department of Aerospace Engineering
And Center for Space Structures and Controls
University of Colorado
Boulder, CO 80309-0429 (U. S. A.)

Abstract

The various forms of parallel numerical algorithms that speed up finite element computations are as different as the number of researchers working on the problem. However, most of the recently proposed concurrent computational strategies stem from the "divide and conquer" paradigm and require domain decomposition or mesh partitioning. In this talk, we present and discuss a family of simple and efficient non-numerical algorithms for the automatic decomposition of arbitrary finite element and finite difference meshes into a specified number of adequately connected and load balanced submeshes. These algorithms cover a wide spectrum of parallel architectures (local memory, shared-memory, massively parallel) and feature a large range of approaches (greedy algorithms, projection methods, bandwidth minimization). In particular we address the issues of communication bandwidth, interface bottleneck, and element recursion. We show how these various decomposers can be applied to implement parallel, massively parallel, and parallel/vector explicit and implicit direct, frontal and iterative finite element and finite difference algorithms, and report on their performance results on the iPSC/2-32, Connection Machine CM2, and CRAY Y-MP. For local memory multiprocessors, each partitioning scheme is augmented with a mapping algorithm which attempts to assign directly connected submeshes to directly interconnected processors.

The proposed mesh decomposers are packaged into a software tool which is hoped to relieve the burden of the preprocessing phase from the methods developers.

References

Abstract: We want to compute the equilibrium positions of hyperelastic bodies under large strain using parallel machines with distributed memory architecture, namely a hypercube. We achieve this goal by using a domain decomposition method at the level of each linearized problem. Assigning one sub-domain per node, the computation and assembly of the local stiffness matrices then become independent tasks, avoiding any communication between the nodes. The remaining computation on the linearized problem involves a mixed solver, meaning that we use both a local direct solver and a global iterative scheme. Namely, a preconditioned conjugate gradient is used at the interface, with a preconditioner that has been chosen in light of keeping the high granularity of the parallelism. The target machine of our experiments is an Intel iPSC/2 hypercube 32SX.

1. The discretized non-linear problem: For the modeling of bodies that can undergo large deformations, we choose the Lagrangian formulation. This means that all variables are defined and maintained in a reference configuration, the main unknown being the displacement field \( u(x) \) of each particle of the domain \( \Omega \) once the loading has been applied.

The equilibrium equation, once discretized on a finite elements basis \(\{\varphi_n\}\), is solved by using a Newton-type method. For compressible materials, the constitutive law takes such a form that each step can be described as:

Step \( n \rightarrow n+1 \)

\[
\begin{align*}
\mathbf{u}^{n+1} &= \mathbf{u}^n - [\mathbf{A}^n]^{-1} \left( \mathbf{G}^n - \mathbf{F}^n \right), \\
\mathbf{A}^n_{\alpha \beta} &= \int_{\Omega} \nabla \varphi_\alpha : \nabla \mathbf{W} \bigg|_{\mathbf{u}^n} : \nabla \varphi_\beta \, dx, \\
\mathbf{G}^n_\alpha &= \int_{\Omega} \nabla \mathbf{W} \bigg|_{\mathbf{u}^n} : \nabla \varphi_\alpha \, dx, \\
\mathbf{H}^n_\alpha &= \int_{\Omega} f \varphi_\alpha \, dx + \int_{\Gamma} g \varphi_\alpha \, ds, \text{ the load,} \\
\mathbf{F}^n(x) &= \mathbf{Id} + \nabla \mathbf{u}(x) \text{ is the deformation gradient,} \\
\text{and } \mathbf{W}(\mathbf{F}) &= \text{the specific internal elastic energy.}
\end{align*}
\]

In turn, for isotropic materials, \( W \) only depends on the invariants of the right Cauchy-Green tensor \( F \times F \):

\[
\begin{align*}
l_1 &= \text{Tr}(F \times F), \\
l_2 &= \text{Tr}((\text{adj}F)^T \times \text{adj}F), \\
J &= \det(F).
\end{align*}
\]

A typical example is given \( W \) the following law for hyperelastic materials:

\[
W(F) = C_1 (l_1 - 3) + C_2 (l_2 - 3) + a (J^3 - 1) - (2C_1 + 4C_2 + 2a) \log J,
\]

where \( C_1, C_2 \) and \( a \) are experimental constants.

To enhance the Newton scheme, we have also applied the following numerical features: the incremental loading and the arc-length continuation, which enable to follow awkward problems like buckling (one of the major needs of large deformations modifications).

2. The mixed linear solver: At each iteration of this process, we must solve a linear system of equations of the kind:

\[
\mathbf{A}^n(U^n) \cdot \mathbf{U}^{n+1} = \mathbf{F}^n(U^n).
\]

The costly step of constructing, assembling and factorizing \( \mathbf{A}^n \) require on a distributed memory machine a coarse-grained parallelism, in order to avoid overwhelming communications. Therefore, the domain decomposition methods achieve this goal by subdividing \( \Omega \) into the geometrical reference domain \( \Omega \) into non-overlapping subdomains \( \Omega_i \), separated by an interface \( \Gamma \). Consequently, the computation and assembly of these subdomain matrices \( \mathbf{A}_i \) become fully independent tasks.

As we could have stated in the previous paragraph, the global stiffness matrix \( \mathbf{A} \) is symmetric, positive, and most often definite (SPD). It is composed from the contributions of the following submatrices:

\[
\begin{align*}
\mathbf{A}_i &= \text{the degrees of freedom internal to } \Omega_i, \\
\mathbf{B} &\text{ interaction between } \Omega_i \text{ and } \Gamma, \\
\mathbf{A}_\Gamma &\text{ contributions due to the elements of } \Omega_i \text{ on } \Gamma.
\end{align*}
\]

By performing Gaussian elimination over the degrees of freedom related to the interior of the subdomains \( \Omega_i \), one obtains a linear system whose unknowns are the degrees of freedom on the interface and whose matrix is the Schur complement matrix \( \mathbf{S} \) on the interface \( \Gamma \). Then, omitting the projections and the mappings, one might observe its useful additive property:

\[
\mathbf{S} = \sum_i \mathbf{A}_i - \mathbf{B}^T \mathbf{A}_\Gamma^{-1} \mathbf{B}.
\]

All components of the local Schur complement matrices can be computed, as in the well known substructuring technique. However, a much less expensive approach consists in keeping an implicit definition of the local Schur complement, and solving the interface problem using a Conjugate Gradient method (CG), because \( \mathbf{S} \) in turn is also SPD. At step \( k \) of the CG, one has to form in parallel the implicit matrix-vector product \( \mathbf{S}_k \mathbf{p} \) where \( p_k \) is the \( k \)th descent direction, which requires the solution of a linear system of matrix \( \mathbf{A}_i \).

Based on an LDL\(^T\) decomposition of \( \mathbf{A}_i \), a direct solver is used in each domain. This factorization is performed once and for all during the initialization step of the inner CG loop.

However, this method requires a preconditioning technique to be efficient. Benefiting from our experiments with the linear three-dimensional elasticity, we have chosen the preconditioner proposed in an analytical form by Glowinski et al in [1]. This amounts to approximating

\[
\mathbf{S}^{-1} = (\sum_i \mathbf{S}_i)^{-1} \text{ by } \mathbf{M}^{-1} = \sum_i \mathbf{D}_i \mathbf{S}_i^{-1} \mathbf{D}_i^T,
\]

where \( \mathbf{D}_i \) are diagonal weighting matrices. This preconditioner is often referred as the Neumann preconditioner, because of the...
analytical boundary conditions described at the interface. Similarly, using the Schur complement amounts to treat a Dirichlet problem regarding the interface.

Without computing the local Schur complement $S$, explicitly, there exists an implicit definition of their inverses:

$$S^{-1} = \begin{pmatrix} I_{r_1} & 0 \\ B & A \\ F & 0 \\ A_1 \\ A_2 \\ A_3 \end{pmatrix}^{-1} \begin{pmatrix} 0 & 1_{r_1} \\ 1 & 0 \\ 0 & 1_{r_2} \\ 1 & 0 \\ 0 & 1_{r_3} \\ 0 & 1_{r_4} \end{pmatrix}.$$

Thus, only an LDL$^T$ factorization of the local stiffness matrices has to be performed, in order to compute products of vectors by the preconditioning matrix $M$.

In the linear framework, we have already extensively studied this method, in [2]. It shows a great potential for parallelization, especially for distributed memory architecture and it proved to be robust in problems arising from anisotropic and non-homogeneous materials. We also have computed theoretical bounds for the condition number of the iteration matrix, in [3], we proved that in a general partitioning, it grows like $O\left(\frac{d}{h^2} \left(1 + \log \left(\frac{d}{h}\right)\right)\right)$, $d$ being the average diameter of a subdomain and $h$ the scale of the mesh. The weak dependency on $h$ means that the preconditioner is still profitable when the mesh is refined. However, the number of subdomains should not increase too much, and a limit of 16 seems reasonable, especially when the partitioning is performed in the three-dimensional space.

Without the optimality of the preconditioner described in [5] by Smith, whose condition number is independent of $d$ and $h$, the "Neumann" preconditioner remains local and easy to construct on a large unstructured mesh: indeed it requires neither the explicit Schur Complement matrices nor any coarse mesh.

This versatility of the "Neumann" preconditioner favors its implementation in non-linear Finite Element problem.

The data structure for the interface: The distributed architecture of the computer must be taken into account in the choice of a suitable data structure for the interface.

At each iteration of the PCG, there are 3 interface vectors to be stored: $u$, $r$, and $p$ the displacement, the residual and the descent direction, respectively; and 2 interface vectors to be used $t = \sum S_p$ and $z = Mr$, the conjugate direction and the preconditioned residual, respectively.

We describe a strategy of implementation, called global interface, where each node of the machine redundantly stores the whole interface, as opposed to the so-called local interface, where each node of the machine only knows the interface degrees of freedom which are in its immediate neighborhood.

This global interface also reduces some redundant computation, but reduces the number of communications. On the one hand, the vectors to be transferred are longer, on the other hand, no communications are needed to perform the dot-products (whereas in the other approach, local weighted subproducts are computed and then gathered).

With this data structure, the only exchanges are the global summations: those are scheduled to be performed by physical directions of links (see Saad [1]), thus no special mapping of the subdomains onto the nodes is required, e.g. by a binary Gray Code. Thanks to the unicity of the structure of the interface, one level of indirect addressing is also suppressed at the gathering operation.

A complex splitting becomes more difficult to handle with the global data structure. However, with no complementary communications, the code has been implemented with the following feature: two nodes are dedicated to each subdomain, one storing and using the Dirichlet solver, the other one taking care of the Neumann solver. Notice that on a distributed memory machine, this approach leads to a good parallelization of the memory management and postpone the risk of lack of memory for bigger problems. Of course, these two solvers are still accessed sequentially, but it implies that for a given problem, fitted to processors in memory requirements, one splits the body into $n$ subdomains, overcoming some difficulties of convergence previously quoted. Examples show that it is a matter of trade off.

The results that will be shown have been computed on the Intel iPS/C2 32SX of ONERA, thanks to a cooperation between the Groupe de Calcul Parallèle of ONERA and the Parallel Algorithms Group at CERFACS.

References


Abstract

We are dealing with integral equations related to Helmholtz equation. They come from scattering problems around a compact object. The usual way of discretizing them leads to a full complex non-hermitian matrix. This drastically limits the size of computable problems because of the limited size of memory on computers.

1 Preconditioning

The algorithm used for resolution is Generalized Conjugate Residual Algorithm. It needs to store all directions of descent. Preconditioning has two advantages. First it reduces the CPU time, the resolution and second it leads to fewer directions to be stored, so that we can test meshes with more points and so higher frequencies.

The preconditioning matrix we will focus on consists of a subpart of the full matrix. More precisely we will only retain coefficients that come from interaction between points near from each other (near means the distance is less than a few wavelengths). We first show some theoretical results. The point is that, in a certain extent, we can explain how that preconditioner acts. Indeed it tends to diminish the highest eigenvalues of the matrix with no precise effect on the lower ones. What is really interesting is that these results remain true when frequency grows to infinity.

We then present two series of numerical issues of that preconditioned algorithm. The first one aims to illustrate the reduction of the number of iterations. We can see in tables 1 and 2 that the most remarkable facts is that the number of iterations is reduced from typically 10 to 1 and that seems not to vary with frequency. Table 1 shows numbers of iterations for the unpreconditioned algorithm for various frequencies. In these tests, the scatterer is a sphere of radius 1, with sound speed equals to 333. The number of degrees of freedom is 1026. The incident wave is a spherical harmonic. In table 2 we can see the number of iterations needed to achieve convergence. The parameter δ is the maximal distance between 2 points.

Table 1: Unpreconditioned GCRA.

<table>
<thead>
<tr>
<th>δ</th>
<th>780Hz</th>
<th>750Hz</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.40m</td>
<td>10^{-4}</td>
<td>10^{-5}</td>
</tr>
<tr>
<td>0.36m</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>0.32m</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>0.28m</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>0.24m</td>
<td>2</td>
<td>30</td>
</tr>
</tbody>
</table>

Table 2: Preconditioned GCRA.

<table>
<thead>
<tr>
<th>δ</th>
<th>700Hz</th>
<th>600Hz</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.40m</td>
<td>10^{-4}</td>
<td>10^{-5}</td>
</tr>
<tr>
<td>0.36m</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>0.32m</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>0.28m</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>0.24m</td>
<td>4</td>
<td>15</td>
</tr>
</tbody>
</table>

whose mutual interaction is taken into account in the preconditioning matrix. The
second series of results presents comparisons between a monoprocessor (CrayII) and a parallel machine (iPSC-2) in terms of CPU time. In table 3 we present tests for two sizes of mesh. The scatterer is the same as before and the incident wave is a plane. One. In that table, resolution time is the sum of the time of assembly of the matrix and the time of the GCR.

Mesh $n^o1$: 168 points and 368 triangles.

Mesh $n^o2$: 1026 points and 2048 triangles.

We can see that a hypercube with 32 nodes is about the fifth of a processor of CRAYII. Next we will study the way of parallelizing some crucial pieces of the code on an iPSC-2. We will see what we can expect in terms of speed-up related to local granularity. We will see that for a dot product we must increase this granularity to maintain efficiency as we increase the number of nodes when for a matrix-vector product it can remain constant if we take care.

In table 4 we present actual times of computation for a matrix-vector product. $Nloc$ is

<table>
<thead>
<tr>
<th>$nloc$</th>
<th>256</th>
<th>128</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$T_{cal}$</td>
<td>$T_{comm}$</td>
</tr>
<tr>
<td>5-cube</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>4-cube</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>3-cube</td>
<td>10063</td>
<td>26</td>
</tr>
<tr>
<td>2-cube</td>
<td>5039</td>
<td>13</td>
</tr>
<tr>
<td>$nloc$</td>
<td>64</td>
<td>26</td>
</tr>
</tbody>
</table>

Table 4. Matrix-vector product (time in ms).

The size of a vector divided by the number of processor used. $T_{cal}$ and $T_{comm}$ are respectively the time due to computation and communications.
Parallel Grid and Multigrid Methods for Distributed Memory Architectures*

Karl Solchenbach

PALLAS GmbH, Hohe Str. 73, D-5300 Bonn

1 Introduction

Algorithms for the numerical solution of PDEs are typically based on grid data structures (either regular or irregular ones). These algorithms are characterized by inherent parallelism and locality: values at different grid points can be calculated simultaneously and the usually iterative calculation of a grid-point value involves only values at certain neighboring points.

Due to these properties grid-based algorithms (like red-black relaxation) can be implemented on distributed memory architectures very efficiently. Grids are decomposed into subgrids (grid partitioning) and each subgrid is connected to a processor. The locality of the grid operator guarantees that the communication between processors is limited.

The implementation of standard multigrid methods is also based on the grid partitioning approach. The parallel efficiency of multigrid, however, is somewhat less than that of the corresponding single grid method. This is mainly due to the high communication/calculation ratio and the short message length on coarse grids. Nevertheless, the numerical efficiency of multigrid outperforms these losses easily.

2 Parallel grid-based applications

The mathematical model of many different supercomputer applications are formulated as (systems of) partial differential equations (PDEs). The discretization of the PDEs most naturally leads to a grid-based formulation of the problem, i.e., grid data structures and grid-based algorithms.

The implementation on distributed memory parallel computers requires

- the parallelization of the existing algorithms or their substitution by new parallel algorithms;
- the distribution of the data structure to the local memory units. The data distribution should try to preserve locality (i.e., minimize communication) and to achieve load balancing.

2.1 Grid data structures

Distribution strategies and tools have been developed for two classes of grid structures:

- Regular grids are characterized by direct addressing of the grid points and a rectangular or cuboid address space. Geometrical neighbors are also logical neighbors.
- Block-structured grids are composed of several regular grids. Each single block shows internally a regular grid structure; the block structure itself, however, is irregular (with certain restrictions).

Meanwhile also codes based on irregular grids (as used by Finite Element methods) and locally refined grids have been implemented on distributed memory parallel computers. Efficient and comfortable tools for these structures, however, have to be developed yet.

2.2 Parallel grid algorithms

A grid algorithm is a (usually iterative) method which calculates the value of a grid function at a grid point as a function of values defined at neighboring points. The iteration (also called relaxation) can be characterized as Jacobi-type (the new iterate at a grid point is calculated using only old neighboring values) or Gauss-Seidel-type (using already calculated new neighboring values). Obviously, Jacobi-type methods are completely parallel since the calculation in each grid point can be performed independently (cf. Figure 1 (a)). If the number of grid points is \( N \) the parallelism is also \( N \).

The parallelism of Gauss-Seidel methods depends on the order in which the grid points are processed. Lexicographic ordering implies that only points on diagonal lines can be calculated in parallel (cf. Figure 1 (b)).

For Gauss-Seidel methods, a far better degree of parallelism, namely \( N/2 \), is obtained by "coloring" the grid points appropriately and processing all points of the same color simultaneously, e.g. the so-called red-black (RB-) relaxation (cf. Figure 1 (c)).

2.3 Grid partitioning

The usual way to implement parallel grid algorithms on a distributed memory system is based on the method of grid partitioning. The computational domain is divided into several subgrids which are assigned to parallel processes.

Each relaxation step can be performed on a subset of interior points of the subgrid (in Figure 2) independently. Calculation of values at interior boundary points (in Figure 2), however, needs the values from neighboring subgrids (processes). Since the processes have no common data space these values somehow have to be made available.

Instead of transferring the values individually at the time they are needed it is more efficient to have copies of neighboring grid points in the local memory of each process (G). Hence, each process contains...
The grid partitioning approach can be extended to block-structured grids in a straightforward manner.

2.4 Multigrid methods

Standard iterative multigrid algorithms process a cycle from the fine to the coarse grids and back to the fine grids sequentially, whereas on each grid level the actual problem is treated in parallel similarly to the parallel single grid algorithms described in the previous sections.

On parallel distributed-memory systems an efficient implementation of multigrid algorithms is not trivial since the performance may degrade due to:

- idle processors on very coarse grids,
- short messages and dominant influence of start-up time,
- bad communication/computation ratio.

The algorithmical and technical details of parallel multigrid algorithms are described in [3].

2.5 Communications library

For grid applications, the explicit programming of the communication can be hidden from the user. In the SUPRENUM project, for example, a library of communication routines has been developed [1] which ensures

- clean and error-free programming,
- easy development of parallel codes,
- portability within the class of distributed memory computers.

Programs can be ported to any of these machines as soon as the communication library has been implemented.

The library supports regular and block-structured grids and is available at the GMD or at the PALLAS GmbH.

3 Performance

3.1 Performance measures

The quantities of interest in evaluating the performance of parallel algorithms are:

- Time $T(N,P)$: time to solve a problem of size $N$ on a multiprocessor system using $P$ nodes,
- speed-up $S(N,P) := T(N,1)/T(N,P)$,

Note that on the MIMD/SIMD architectures the utilization of the hardware capabilities is the product of the "multiprocessor" efficiency as defined above and the efficiency related to the vector processing unit. The total problem solving time, which is the only interesting number from the user's point of view, depends, of course, additionally on the numerical efficiency of an algorithm.

In practice $E$ will be smaller than its ideal value 1, mainly because of communication (including synchronization), unbalanced load, and sequential parts in the algorithm.

3.2 Performance model for grid applications

Let $N$ be the number of grid-points, $D$ the dimension of the grid (typically $2 \leq D \leq 4$) and $n = N/P$ the size of the local grid on each processor. The total grid is assumed to be cubic.

Then the time for the arithmetic calculations is $T_{\text{arithmetic}} = c_1 n$. On homogeneous parallel architectures the time needed for communication consists of two components, the so-called start-up time needed for the initialization of a message, and the transfer time which is proportional to the length of the message.

The data volume to be communicated depends on the way how the grid is partitioned. If the partitioning is performed in all $D$ dimensions and cubic subgrids are generated, the communication time is $T_{\text{comm}} = c_2 n^{(D-1)/D} + c_3$. The speed-up is

$$S(N,P) = \frac{P}{1 + \frac{c_2}{c_3} n^{-1/D}}.$$ 

The asymptotic behaviour is

$$S(N,P) \rightarrow P, \quad E(N,P) \rightarrow 1 \quad \text{for} \quad n \rightarrow \infty.$$ 

The efficiency remains constant for scaled problems ($n$ constant, $P \rightarrow \infty$).

The constants $c_i$ depend on the particular grid algorithm and the properties of the hardware (communication speed, floating-point performance etc.). A detailed analysis can be found in [2].

References


IMACS session:
Solution of P.D.E. on Massively Parallel
MIMD Systems

P. LECA
ONERA, Parallel Computing Division
B.P. 72, 92322 Chatillon Cedex,
France

Abstract
The success of highly parallel distributed memory multiprocessors will depend mainly on their efficiency when running realistic application codes. This paper presents the main topics discussed in the session dedicated to the use of massively parallel MIMD systems in the field of scientific computing.

In the race to Teraflops performances a new generation of highly parallel multiprocessors is emerging, which is based on the use of a large set of powerful microprocessors. Nevertheless the acceptance of such systems in the industry will depend mainly on their actual performance when running realistic application codes.

This subject has been studied at the University of Colorado where several mesh decomposers has been developed and adapted to various machines such as the hypercube iPSC2, the CRAY multiprocessors and the Connection Machine.

Furthermore representatives from the centers of NASA Ames and NASA ICASE present their last results about the utilization of massively parallel computers for CFD. The respective advantages of SIMD and MIMD computers are particularly discussed.

Then, the experience done at CERFACS concerning the development of a domain decomposition method is presented. This method, that provides a coarse grain parallelism, has been implemented with success on the INTEL iPSC2.

The multigrid technique is now widely used for accelerating the convergence of iterative algorithms on structured grid. However this technique leads to non-local memory references that could enter in contradiction with an efficient implementation on a distributed memory architecture. The experience gathered at SUPRENUM on this subject is also addressed in this session.

At last, dealing with very large dense matrices issued from an integral equation formulation, recent algorithms developments done at ONERA adequate to the numerical computation of Helmholtz equation are presented with implementation results on CRAY-2 and iPSC2 multiprocessors.

This session focuses on the development and the implementation of these methods on massively parallel MIMD architectures (iPSC, NCUBE, BBN ...).

On such systems the efficiency is often the result of a trade-off between the reduction of the time due to data communication, either in the communication network or through the memory hierarchy, and the reduction of the computation time. Then parallel algorithms exploiting data locality in private or local memory are specially stressed.

Moreover, the use of a geometric parallelism, based on a partition of the computational domain, pushes the development of software tools that provide automatically this partitioning and the corresponding data structures.
PARALLEL SYNERGY:
CAN A PARALLEL COMPUTER BE MORE EFFICIENT THAN THE SUM OF ITS PARTS?
Selin G. Akl
IEEE Senior Member
Department of Computing and Information Science
Queen’s University, Kingston, Ontario K7L 3N6
CANADA

Abstract: The two most popular models of sequential and parallel computation
lead, once defined precisely, to a computational paradox. Specifically, we show
that for a wide family of problems the cost of a PRAM solution is smaller than
that of its RAM counterpart. This contradicts the currently established belief,
and does not appear to be amenable to explanation using existing approaches. We use
the term parallel synergy to refer to this phenomenon.

1. INTRODUCTION

Over the last forty years the Random Access Machine (RAM) has established
itself as the most widely understood and used model of sequential computation
(Engeler, Hopcroft, Machtey, Mandrioli, etc.). Here several processors share a common random-access memory
and access simultaneously to the same memory location (for reading or for writing). In
variants depending on whether two or more processors
of a given
phase of an algorithm. Note that some processors may not execute
during a READ or WRITE every processor
may access a PA simultaneous
The Parallel Random Access Machine (PRAM) appears to
be the preferred model among theoretical computer scientists
When we refer to the cost of a network for memory access, indeed, both the
and the PRAM ignore that network's expense since it is just the-
circuit's cost. We use the term cost criterion (not to
the cost analysis.
In parallel computation, the Parallel Random Access Machine (PRAM) appears to
be the preferred model among theoretical computer scientists [Akl 1, Gibbons, Karp, Papert]. Here several processors share a common random-access memory. As in the RAM, each processor has a constant number of local registers. All processors simultaneously execute the instructions of a parallel algorithm. Each step of such an algorithm consists of (up to) three phases:

(i) a READ phase, where the processor reads a datum from the random access memory and stores it in one of its local registers;

(ii) a COMPUTE phase, where the processor performs an elementary operation (such as comparison, addition, etc...) on data in its local registers;

(iii) a WRITE phase, where the processor writes into the random access memory the result of some computation.

Each of these phases is assumed to take constant time, leading to a constant execution time per step. It is important to emphasize that the model as described assumes that each of the operands and results of an elementary operation fits into a single memory location. Thus, in the terminology of [Akl 1], we are using the "uniform cost criterion" (not to be confused with the cost of a computation defined below).

In parallel computation, the Parallel Random Access Machine (PRAM) appears to
be the preferred model among theoretical computer scientists [Akl 1, Gibbons, Karp, Papert]. Here several processors share a common random-access memory. As in the RAM, each processor has a constant number of local registers. All processors simultaneously execute the instructions of a parallel algorithm. Each step of such an algorithm consists of (up to) three phases:

(i) a READ phase, where processors read data from the shared memory and store them in their local registers;

(ii) a COMPUTE phase, where processors perform elementary operations on
local data;

(iii) a WRITE phase, where processors write results to the shared memory.

Each of these phases requires constant time, again leading to a constant time per
step of the algorithm. Note that some processors may not execute one or two phases of a given step. Also, during a READ or WRITE, every processor may gain access to a different memory location. However, the PRAM gives rise to a number of variants depending on whether two or more processors are allowed to gain
access simultaneously to the same memory location (for reading or for writing). In
this paper we shall be concerned solely with that variant of the PRAM which
disallows such simultaneous access to the same memory location. This model is
known in the literature as the Exclusive-Read Exclusive-Write (EREW) PRAM.

We define the cost of an algorithm as being the product of the number of processors it uses and its running time [Akl 1, Quinn]. [Note that some authors use the term work instead of cost [Comer].] It is usually said that if c is the cost of running an algorithm A on a RAM, then the cost of simulating A on a RAM is (asymptotically) equal to c [Akl 1, Almasi, Eager, Faber 1, Faber 2, Fishburn, Moli]. We argue in this paper that this statement is no longer true once the network interconnecting processors to memory locations is taken into consideration by the cost analysis. More precisely, we show that for a wide family of computational problems, the cost of a PRAM solution is smaller (asymptotically) than that of the best possible sequential solution. Furthermore, the cost of the PRAM solution grows when simulated on a RAM. We emphasize here that the question addressed in this paper (namely inclusion of the interconnection network in the cost analysis) is distinct from the problem treated in [Akl 3, Gmi, Gustafson, Janssen, Komlfi, La, Leach, Li 1, Li 2, Melhorn, Parkinson, Press, Quinn, Wesel], where various issues pertaining to speedup (i.e. the ratio of sequential to parallel running time) are discussed.

2. TRADITIONAL COST ANALYSIS

Assume that a problem P of size n is given, for which \( T_s(n) \) is a sequential algorithm running in sequential time \( T_s(n) \) on a RAM. The cost of \( T_s(n) \) is \( c_s(n) = 1 \times T_s(n) = T_s(n) \). Note that, because there is only one processor, the cost of a sequential algorithm is exactly its running time.

Further, let \( T_p(n) \) be a parallel algorithm for P running in parallel time \( T_p(n) = T_s(n)/n \) on a PRAM with n processors. The cost of \( T_p(n) \) is \( c_p(n) = n \times T_p(n) = T_s(n) \).

As \( c_s(n) \) and \( c_p(n) \) are derived for the RAM and PRAM, respectively, they do not
take into consideration the network required for memory access. Indeed, both the
PRAM and the RAM ignore that network's expense since it is just the circuit's
cost. We use the term cost criterion (not to be confused with the cost of a computation defined below).

We note in passing that an interconnection network that does not allow feedback,
. i.e. a network where each processor is used once per memory access, is sometimes referred to as a circuit (Papert). Occasionally, the cost (or size) of a circuit is expressed simply as the number of processors it uses, without multiplying the latter by the time it takes to traverse the circuit (also known as the circuit's depth).

In this paper, we prefer to adhere to the standard definition of cost (namely, number of processors x running time), in order to avoid restricting our discussion to circuits. As it turns out, inclusion of the memory access time in the analysis affects our results only marginally.

3. TRUE EFFICIENCY

In the RAM, in order to gain access to any of n memory locations, the processor
issues a log n - bit address. A network of size O(n) decodes this address in O(\log n) time. This network is, for example, a binary tree of processors (i.e. a circuit) in the terminology defined earlier) [Kuck, Tanenbaum]. It has a cost of O(n \log n).

In the EREW PRAM, processors gain access to memory locations simultaneously (one memory location per processor) in O(\log^2 n) time using a network of O(n \log^2 n) processors where \( a \) and \( b \) are two constants. Typical values of \( a \) and \( b \) are given in Table 1, along with references to the corresponding networks.

The cost of the network for memory access is therefore O(n \log^2 n), where \( d = a + b \).

<table>
<thead>
<tr>
<th>Network</th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Batcher]</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>[Stone]</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>[Ajtai]</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>[Leighton]</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1. Values of \( a \) and \( b \) for typical networks used in memory access.
The revised cost of \( A_p^2 \) is therefore,
\[
c'(p)(n) = (1 + O(n)) \cdot (1 + O(n) \cdot O(\log n) = (1 + O(n) \cdot O(n \log n).
\]
while the revised cost of \( A_p \) is,
\[
c'(p)(n) = (1 + O(\log n)) \cdot (1 + O(n)) \cdot O(n \log n) = (1 + O(n) \cdot O(\log d n).
\]

In other words, the cost of the parallel algorithm is asymptotically smaller than that of the sequential algorithm. Notice that, because \( x_{p,n} \) is defined as \( \log n \), the exact value of \( x_{p,n} \) is unclear when computing the ratio \( c'(p)/(n) \cdot O(n) \). If so needed, one may of course assume that \( c'(p) \) is the running time of the fastest possible algorithm for \( p \).

It should also be clear that our analysis holds for any number of traditional PRAM processors, provided that this number is at most a linear function of \( n \), the number of PRAM memory locations required to solve a given problem of size \( n \). Indeed, denoting the number of PRAM processors by \( N \), where \( N = n \), we see that the asymptotic value of \( c'(p)/O(n) \) remains unchanged, namely,
\[
c'(p)(n) = (1 + O(n) + O(n)) \cdot (1 + O(n)) \cdot O(n \log n(n + N))
\]
\[
= (1 + O(n)) \cdot (1 + O(n) \log n).
\]

As noted above, the traditional approach to analyzing parallel algorithms does not take into consideration the cost of the network interconnecting processors to memories. In this approach, the efficiency of a parallel algorithm for a given problem is defined as the ratio of the running time of the fastest sequential algorithm for that problem to the cost of the parallel algorithm. Because of the assumption that a RAM can simulate a PRAM algorithm in no more time than the cost of the latter, the efficiency of a parallel algorithm is at most 1. By contrast, we refer in this paper to the ratio \( c'(p)/O(n) \) as the true efficiency of a parallel algorithm.

This ratio in the case of \( A_p \) and \( A_p^2 \) of \( O(n) \), which is larger than 1. In what follows, we assume that the circuit of (Ajitai) is used to implement the EREW RAM, i.e., \( d = 2 \).

### 4. EXAMPLE

Consider the following problem. We are given \( n \) distinct integers \( I_1, I_2, \ldots, I_n \) in the range \(-m, m \), stored in an array \( X_1, X_2, \ldots, X_{2n} \) such that \( X_{2i} = I_i \) for all \( 1 \leq i \leq n \). It is required to modify \( X \) so that \( X_{2i} = I_i \) for all \( 1 \leq i \leq n \), if and only if \( 1 \leq i \leq n \), otherwise \( X_{2i} = I_i \) [Akitai 3].

Sequentially, the problem can be solved on the RAM in the obvious way in time \( O(n) = O(n) \), and this is optimal since every entry of \( X \) must be examined once. Consequently, \( c'(p) = O(n \log n) \). In parallel, \( c'(p) = O(\log n) \) on a PRAM with \( n \) processors. Thus, \( c'(p) = O(n \log n) \). It follows that \( c'(p)/O(n) = O(n / \log n) \).

Note that simulating the PRAM algorithm on the RAM leads to an algorithm whose cost is
\[
O(n) \cdot c'(p)/O(n) = O(n^2 \log n).
\]

This cost is larger than the (optimal) costs of both the RAM and PRAM solutions.

### 5. GENERALIZATION

In general, any computational problem of size \( n \), \( c'(p)/O(n) > 1 \), i.e., \( t(p)/O(n) = n \log n \), \( c'(p)/O(n) = n \log n \), provided that \( t(p) \rightarrow O(n) \), and the number of steps where a READ and/or a WRITE phase is executed dominates the computation. This condition is satisfied by a wide family of problems. These problems include selection, merging, sorting, and a variety of computations in numerical analysis, graph theory, and computational geometry [Akitai 1].

### 6. DISCUSSION

Traditional analyses of cost either ignore the existence of a network to interconnect processors to memory locations, or (implicitly) assume that the cost of such a network is \( O(1) \). Both approaches are clearly unrealistic; any reasonable model of computation must include an integral part of the problem domain, such as a network of linked processors to memory locations, whose cost is a function of the number of these processors and memory locations.

On the other hand, in deriving \( c'(p) = \log n \) and \( c'(p) = \log d n \), we have taken into account both the number of processors required for such a network and the cost of the network. It may be argued that since the processors used to build the interconnection network are simpler than those actually doing the arithmetic and logical computations, the two ought not be treated equally in the cost analysis. The fallacy in this argument is that the cost of both is constant. The traditional approach to analyzing parallel algorithms is thus circular: the cost is considered to be the sum of the cost of the network multiplied by the cost of the parallel algorithm. In that approach, the cost of the network is considered to be the cost of the algorithm.

### 7. CONCLUSION

From the above discussion we conclude that the PRAM allows for a synergistic phenomenon to occur. This phenomenon manifests itself by a reduction in computational cost (as defined in this paper) for a wide variety of problems. These reductions are characterized by an exchange of work and time between the RAM and the PRAM. A PRAM is a powerful model of computation, as it provides a parallel model of computation, which is essentially equivalent to the sequential model. However, the cost of a parallel algorithm is not negligible. The PRAM is, in fact, more powerful than the RAM, and the cost of a parallel algorithm is greater than the cost of the sequential algorithm. The cost of the parallel algorithm is thus smaller than the cost of the sequential algorithm.

This result has a number of implications. Our work was originally motivated by the observation that both the RAM and the PRAM are models of parallel computation, which are essentially equivalent. In the parallel model, the cost of the algorithm is negligible. However, the RAM is a more powerful model, and the cost of the algorithm is not negligible.

### 8. REFERENCES


RELATIONAL STRUCTURE SEMANTICS OF CONCURRENT SYSTEMS

Ryszard Janicki
Department of Computer Science and Systems
McMaster University
Hamilton, Ontario, Canada, L8S 4K1
email: janicki@ca.mcmaster.des.maces

Maciej Koutny
Computing Laboratory
The University of Newcastle upon Tyne
Newcastle upon Tyne NE1 7RU, U.K.
email: Maciej.Koutny@uk.ac.newcastle

Abstract We introduce a new relational structure semantics of concurrent systems which is a generalisation of the causal partial order semantics. The new semantics is consistent with the operational behaviour of inhibitor and priority nets expressed in terms of step sequences. It employs combined partial orders - composites (each composite is a relational structure consisting of a causal partial order and a weak causal partial order). We outline the way in which composites can be generated for 1-safe inhibitor nets.

1. INTRODUCTION

In the development of mathematical models for concurrent systems, the concepts of partial and total order undoubtedly occupy a central position. Interleaving models use total orders of event occurrences, while so-called 'true concurrency' models use step-sequences or causal partial orders (comp. [BD87, Ho88, Pr86]). Even more complex structures, such as failures [Ho88] or event structures [Wi82], are in principle based on the concept of a total or partial order. While interleavings and step sequences usually represent executions or observations and can be regarded as directly representing operational behaviour of a concurrent system, the causality relation represents a set of executions or observations. The lack of ordering between two event occurrences in the case of a step sequence is interpreted as simultaneity, while in the case of a causality relation it is interpreted as independency, which means that the event occurrences can be executed (observed) in either order or simultaneously. In other words, a causal partial order is an invariant describing an abstract history of a concurrent system. Both interleaving and true concurrency models have been developed to a high degree of sophistication and proved to be successful specification, verification and property proving tools. However, there are some problems, for instance the specification of priorities using partial orders alone is rather problematic (see [La85, Ja87, JL88]). Another example are inhibitor nets (see [Pe81]) which are virtually admired by practitioners, and almost completely rejected by theoreticians, in our opinion mainly because their concurrent behaviour cannot be properly defined in terms of causality based structures. We think that these kind of problems follows from the general assumption that all behavioural properties of a concurrent system can be adequately modelled in terms of causal partial orders or causality-based relations. We claim that the structure of concurrency phenomena is richer and there are other invariants which can represent an abstract history of a concurrent system. In this paper we will show how one of such invariants (weak causal partial order) can be defined and derived. Our main result will be the definition of a relational structure semantics for inhibitor nets which employs combined partial orders composites to model the interrelationship between event occurrences involved in a concurrent history (each composite is a relational structure consisting of a causal partial order and a weak causal partial order). The resulting semantics model is consistent with the operational behaviour of such nets expressed in terms of step sequences. We also show that the way in which composites can be generated is a direct generalisation of the procedure used to generate causal partial orders. The results of this paper are directly applicable to systems with priorities, nets with inhibitor arcs, bounded nets, and virtually any system model which supports the notion of true concurrency semantics.

2. MOTIVATION

In this section we present an example which we believe clearly identifies an inability of the causal partial order (CPO) semantics to cope properly with some important aspects of non-sequential behaviour. We will use Petri nets [Pe81, Re85] to illustrate our discussion.

Our example closely follows the discussion in [Ja87, JL88]. We consider a concurrent system Con comprising two sequential subsystems A and B such that:

1. A can execute event a and after event b.
2. B can either engage in event b, or engage in event c.
3. The two sequential subsystems synchronise by means of the handshake communication.
4. The specification of Con includes a priority constraint stating that whenever it is possible to execute event b, then event c must not be executed.

One can model Con as the Petri net Nprior in Figure 1. Before analysing the behaviour of Nprior, we look at the behaviour of net N, where N is Nprior without the priority con-

![Image](https://example.com/image.png)

**Figure 1**
constraint. The operational behaviour of $N$ can be captured by the set of step sequences it generates, each step being a set of events executed simultaneously, as follows:

$$\text{steps}(N) = \{ \lambda, \{a\}, \{c\}, \{a\}{b}, \{a\}{c}, \{a, c\}, \{c\}{a} \}.$$ 

Note that $\lambda$ denotes the empty step sequence. A fundamental result of the theory of causal partial orders now says that there is a set of partially ordered sets, posets($N$), such that

$$\text{steps}(N) = \bigcup_{\text{posets}(N)} \text{steps}(p)$$

where steps($p$) is the set of all step sequences consistent with a poset $p$. Figure 2 shows the elements of posets($N$) together with the sets steps($p$). Note that each poset posets($N$) is interpreted as a causality relationship (an invariant) involving event occurrences, and is intended to represent an abstract history of net $N$. The consistency between the operational and invariant semantics captured by (2.1) is a cornerstone of the theory of causal partial orders.

Having looked at the two-level (i.e. operational and invariant) description of the behaviour of $N$, one might attempt to repeat the same construction for the priority net $N_{\text{prior}}$. It is relatively easy to obtain the operational semantics of $N_{\text{prior}}$. We simply delete from steps($N$) those step sequences which are inconsistent with the priority constraint. As the result we obtain:

$$\text{steps}(N_{\text{prior}}) = \{ \lambda, \{a\}, \{c\}, \{a\}{b}, \{a, c\}, \{c\}{a} \}.$$

Note that we deleted $\{a\}{c}$ since after executing $a$, event $b$ becomes enabled and thus $c$ cannot be executed. We should now be able to find a set posets($N_{\text{prior}}$) such that

$$\text{steps}(N_{\text{prior}}) = \bigcup_{\text{posets}(N_{\text{prior}})} \text{steps}(p).$$

However, any attempt at finding such a set has to fail.

Proposition 1

There exists no set of partially ordered sets posets($N_{\text{prior}}$) such that (2.2) holds.

This leads us to a conclusion that it is impossible to construct a CPO semantics of the priority net $N_{\text{prior}}$ which would be consistent with the full operational behaviour of that net, steps($N_{\text{prior}}$). In order to develop an invariant semantics for $N_{\text{prior}}$ which would be consistent with steps($N_{\text{prior}}$), we must go beyond the CPO-based framework. In the rest of this paper we will show that there is a relational structure semantics of $N_{\text{prior}}$, called combined partial order semantics - composets($N_{\text{prior}}$) - such that

$$\text{steps}(N_{\text{prior}}) = \bigcup_{\text{composets}(N_{\text{prior}})} \text{steps}(c).$$

We will show that the composet semantics is a generalisation of the CPO semantics, and that the way it may be derived is very similar to that used to derive the standard CPO semantics. We finally note that in this paper by a partially ordered set (poset) we mean a pair $(X,R)$ such that $X$ is a set, and $R \subseteq X \times X$ is irreflexive ($\neg aRa$) and transitive $(aRb \land bRc \Rightarrow aRc)$.

### 3. AN OUTLINE OF THE COMPOSET MODEL

In this and the following sections by a step sequence we will mean a special kind of labelled partial order (for which the un-ordering relation is transitive) rather than a sequence of events. The two representations are isomorphic, and the translation from the latter to the former is illustrated in Figure 3 (see [Ja87] for details).

If we look closer at the CPO model, it turns out that an abstract history $H$ of a concurrent system can be represented in either of the following two forms:

(3.1) On the invariant level - as a poset which captures the causality relationship between the event occurrences involved in history $H$.

(3.2) On the operational level - as a set of step sequences being underlain by the same causal relationship, i.e., as steps($p$) for some $p$ from (3.1).

In addition, we have some properties which establish the consistency between these two different views on $H$.

(3.3) Each poset $p$ can be realised on the operational level, i.e., steps($p$) $\neq \emptyset$.

(3.4) Each poset is uniquely identified by the step sequences which are consistent with it, i.e., steps($p$) $=$ steps($p'$) implies $p$ $=$ $p'$.

(3.5) Each poset $p$ is indeed an invariant, i.e., $a$ precedes $b$ in $p$ if and only if $a$ precedes $b$ in every step sequence in steps($p$).

The main reason why posets are adequate representations of concurrent histories in the CPO model is that the whole approach is founded upon the following assumption concerning the relative order of two event occurrences, $a$ and $b$, involved in a concurrent history $H$.

(3.6) The existence of a step sequence in $H$ in which $a$ and $b$ occur simultaneously is equivalent to the existence of two step sequences in $H$, one in which $a$ precedes $b$, the other in which $a$ follows $b$.

From (3.6) it follows that a poset can be an adequate representation of a concurrent history $H$. There are exactly three possible invariant relationships between two event occurrences, $a$ and $b$, involved in $H$.
4. THE MODEL

We define a composet (or combined partially ordered set) as a relational structure (see [Co81]) \( \Sigma = (\Sigma, \rightarrow, \succsim) \) such that \( \Sigma \) is a finite set of event occurrences and \( \rightarrow, \succsim \) are relations on \( \Sigma \) satisfying the following.

1. \( \neg \rightarrow a \)
2. \( a \succsim b \Rightarrow \neg b \rightarrow a \)
3. \( a \rightarrow b \Rightarrow a \succsim b \)
4. \( a \land b \land b \land c \Rightarrow a = c \lor a \land c \)
5. \( a \land b \land b \land c \Rightarrow a \rightarrow c \)
6. \( a \rightarrow b \land b \land c \Rightarrow a \rightarrow c \).

Corollary 2

\((\Sigma, \rightarrow)\) is a partially ordered set, while \((\Sigma, \succsim)\) is a pre-ordered set. \( \square \)

The two relations constituting a composet can be derived as invariants of a set of step sequences with the same domain.

Proposition 3

Let \( \Delta \) be a non-empty set of step sequences with a common domain \( \Sigma \). Let \( \rightarrow_\Delta \) and \( \succsim_\Delta \) be two relations on \( \Sigma \) defined by

\[
\begin{align*}
\rightarrow_\Delta & : \forall \alpha \in \Delta. \ a \rightarrow \beta \\
\succsim_\Delta & : \forall \alpha \Delta. \ a \succsim \beta \lor a \rightarrow \beta
\end{align*}
\]

where \( a \rightarrow \beta \) and \( a \succsim \beta \) respectively mean that \( a \) precedes \( b \) and \( a \) is simultaneous with \( b \) in \( \alpha \).

Then \( \alpha = (\Sigma, \rightarrow_\Delta, \succsim_\Delta) \) is a composet. \( \square \)

I.e., a composet can in a natural way be derived as an invariant of a set of step sequences. To show that a composet is an adequate invariant representation we need another result.

Let \( \alpha = (\Sigma, \rightarrow, \succsim) \) be a composet. The set of step sequences consistent with \( \alpha \), \( \text{steps}(\alpha) \), comprises all step sequences \( \alpha \) with the domain \( \Sigma \) satisfying the following.

1. \( a \rightarrow b \Rightarrow a \rightarrow \beta \)
2. \( a \succsim b \lor a \rightarrow \beta \).

Then the adequacy of the composet notion follows from the following result (comp. (3.5)).

Theorem 4

\( \text{If } \alpha \text{ is a composet and } \Delta = \text{steps}(\alpha) \text{ then } \alpha = (\Sigma, \rightarrow_\Delta, \succsim_\Delta) \) \( \square \)

The next result is a direct generalisation of the properties of the CPO model captured by (3.3) and (3.4).

Theorem 5

1. For every composet \( \alpha \), \( \text{steps}(\alpha) = \emptyset \).
2. If \( \text{steps}(\alpha) = \text{steps}(\alpha') \) then \( \alpha = \alpha' \). \( \square \)

The composets together with the steps operation can provide an invariant model in exactly the same way as the causal partial orders. To show that the new model overcomes the shortcomings of the CPO model, we now have the following.

Proposition 6

There is a set of composets, \( \text{composets}(N_{\text{prior}}) \), such that (2.3) holds. \( \square \)

In this way we have solved the problem from Section 2, i.e., we have found an invariant semantics of \( N_{\text{prior}} \) which in a direct way generalises the CPO semantics (note that each poset \((\Sigma, \rightarrow)\) is isomorphic to the composet \((\Sigma, \rightarrow, \succsim)\)), and is consistent with the full operational semantics of \( N_{\text{prior}} \). Note that the abstract histories of \( N_{\text{prior}} \) may be represented either as composets or as sets of step sequences which are consistent with those composets.

There is a certain similarity between our definition of the composet and the axioms for strong and weak precedence relation presented in [La86]. However, the way these two concepts are derived, the motivations, and the reasons for their introduction are quite different. Hence this similarity is either accidental or, as we would suggest, the composet is a
natural generalisation of the concept of the partial order, and it may be useful for various, perhaps unrelated, applications.

5. THE CONSTRUCTION OF COMPOSETS

In this section we outline an algorithm for constructing the set of composes of a concurrent system. Since in our construction we can use a number of notions which have been developed for ordinary Petri nets, we will show the construction for 1-safe inhibitor nets [Pe81]. Note that 1-safeness means that each place may hold at most one token, and an inhibitor net is a Petri net with added inhibitor arcs. An inhibitor arc between place \( p \) and transition (event) \( t \) means that \( t \) can be enabled only if \( p \) is not marked. In the diagrams an inhibitor arc is identified by a small circle at one end.

The standard approach in which the CPO semantics for ordinary 1-safe Petri nets is derived employs occurrence nets (see [Re85, BD87]). An occurrence net is a representation of the causality relation on event occurrences (or single abstract history of the net). It is an unmarked acyclic net whose each place has at most one input and one output transition. Occurrence nets can be obtained by unfolding marked nets and resolving the conflicts according to the firing rules. This is illustrated in Figure 4(a,b). Each occurrence net induces a poset on event occurrences in the following way: First an auxiliary relation \( \rightarrow_{\text{aux}} \) is derived by transforming each three-node path \( \text{event1} \rightarrow \text{place} \rightarrow \text{event2} \) in the graph of the occurrence net into a pair \( \text{event1} \rightarrow_{\text{aux}} \text{event2} \). Then a CPO is obtained by taking the transitive closure of \( \rightarrow_{\text{aux}} \). For the occurrence net of Figure 4(b), the auxiliary relation \( \rightarrow_{\text{aux}} \) is shown in Figure 4(c), and the resulting causal partial order is shown in Figure 4(d).

The way in which we construct composes for an inhibitor net will closely follow the above procedure. Let \( N_I \) be the inhibitor net shown in Figure 5. We first define an occurrence net of an inhibitor net by generalising in a straightforward way the standard definition of an occurrence net of an ordinary Petri net. The only new element is the handling of the inhibitor arcs. Since in the occurrence net places represent tokens, it is not possible to join \( c \) with place \( 2 \) using an inhibitor arc. However, we can join \( c \) with the complement place [Re85] of \( 2 \), i.e., place \( 5 \), using an activator arc (with a black dot at one end). Intuitively, this means that \( c \) can be executed only when \( 5 \) is marked. We also note that there is no restriction on the number of activator arcs which can be adjacent to a single place. A possible occurrence net for the inhibitor net \( N_I \) is shown in Figure 6.

The next step is to transform the structural relationships embedded in the graph of the occurrence net into two auxiliary relations, \( \rightarrow_{\text{aux}} \) and \( \rightarrow_{\text{aux}} \), from which the composite can be derived. There are three structural relationships between event occurrences which we need to consider, as shown in Figure 7. For the occurrence net of Figure 6 the two auxiliary relations are shown in Figure 8(a). The final step has to take into account the various transitivities from the definition of a composite. More precisely, if \( \rightarrow_{\text{aux}} \) and \( \rightarrow_{\text{aux}} \) have been defined for an occurrence net \( ON \) with \( \Sigma \) being the set of event occurrences, then the composite induced by \( ON \), is defined as \( \text{co}(ON) = (\Sigma, \rightarrow, \rightarrow) \), where \( (\Sigma, \rightarrow, \rightarrow) \) is a minimal composite (w.r.t. set inclusion for both \( \rightarrow \) and \( \rightarrow \)) such that...
Abstract. A 5-state solution of the problem of generating the stable segment of length \( L \) with 1-D scope-3 cellular automaton is presented. \( L \) is represented by the initial configuration in binary form of some polynomial \( R(L) \). Here \( L > 0 \) and simply \( R(L) = L \). Carry-free counter with states in RNS form and two waves transmitting the information were used in the solution. The generation needs \( T = 2L + 3 \) steps of time. Further, it is shown how \( R(L) \) is related to the speed of the front wave. A comparison of some solutions is also given.

I. INTRODUCTION.

We focus here on generating a stable 1-D final pattern, called the segment and composed of \( L \) consecutive states \( x \) starting from an initial configuration which contains certain encoded information about \( L \). The problem is to design a minimal state 1D scope-3 cellular automaton (CA) capable to realize such transformation. This will be referred to as CSG (cellular segment generation) problem. The problem was posed and partially solved in 1975 [1]. The immediate application of the CSG arise in VLSI circuits when \( L \) is assumed position \( \gamma \). An effort was undertaken on looking for a firing squad problems were We shall represent finite configurations \( S_0 \) such a configuration \( X \) of \( X \) starting from an any position \( i \). The alphabet and \( X \) is determined by \( f(P(c)) \). Consequently for each \( i \) to \( i \) \( X \) we have \( \nu (c_i) = (c_{i - 1}, c_i, c_{i + 1}) \) with left and right neighbors. The input alphabet \( S_0 \) of \( c_i \) is determined by the states of its neighbors.

An alphabet(tuple \( \{a, b, c, d\} \) of elements from set \( S \) for a given CA, such that \( \lambda(a, b, c, d) = d \) will be called the elementary rule (ER) of CA and will be denoted by \( abc/d \); here the term "production" was also in use [4]. The local function \( \lambda \) of CA may be described then by the set of all its ERs. A state denoted by \( '-' \) will be called the quiescent state with a property that every is the ER of CA. A configuration of \( CA = (C, \lambda, \lambda) \) is defined as the function \( f. C \rightarrow S \) so \( S \) denotes the set of all configurations. Then, \( f(\nu(c)) \) is the state of neighborhood of given cell \( c \). The configuration \( f \) is called finite if and only if \( f(c_i) = . \) for all but finitely many \( i \)’s.

We shall represent finite configurations as follows. If \( f \in S^C \) is such that for some \( i \) the equality \( f(c_{i+j}) \) holds for all \( j < 0 \) and for all \( j > k - 1 \) then \( f \) will be denoted by \( \cdots \cdot w \cdots \cdot \) where \( w = f(c_j) \cdots f(c_{j+k-1}) \) and \( k > 0 \). The configuration \( f \) will be said to have the length \( k \) and to occur at the position \( i \).

Suppose now that binary digits 0 and 1 are in state set \( S \). Let \( R(L) = \alpha L + \beta \) be a linear polynomial with \( R(L) > 0 \) for all integers \( L \). Assume some \( L > 0 \) and \( \omega = (R(L))^2 \) where the binary form of \( R(L) \) has \( r \) digits with 1 as the leftmost digit. Then the configuration
... will be called either the simple representation of L (if \( a = 1 \) and \( b = 0 \)) or the encoded representation of L, otherwise.

Let \( x \in S \) and let \( \cdots(x)_L \cdots \) denotes a configuration of L consecutive cells, all occurring in state x. It will be called the segment of length \( L \).

By global transition function \( \gamma : S^C \rightarrow S^C \) of CA such a function is understood that \( \gamma(f) = \varphi \) implies \( \gamma(f(e_L)) = \lambda(f(\varphi(e_L))) = \varphi(e_L) \) for all \( 0 \leq i < L \). Then we say that CA passes from \( f \) to \( \varphi \).

Let \( f_0 = \cdots(x)_L \cdots \), be some representation of given integer \( L > 0 \). If CA with global transition function \( \gamma \) passes from given initial \( f_0 \) to the final stable configuration \( f_T = \cdots(x)_L \cdots \) in such a way that:

\[
\gamma(f_0) = f_1, \quad \gamma(f_1) = f_2, \quad \ldots \quad \gamma(f_{T-1}) = f_T
\]

where \( f_{i-1} \neq f_i \) for all \( i = 1, 2, \ldots, T \), then we say that CA generates the segment of length \( L \). \( T \) is the time of CA process.

Now we can state the CA problem: one may determine 'simple' CA = \( C(V, \lambda, \lambda) \) capable to generate segments for any given length \( L > 0 \). Since \( \lambda = 3 \) has been chosen, then \( S \) and \( \lambda \) rest to be searched for in the problem.

### III. SOLUTION

The solution presented here for the CA behaviour assumes two levels of organization. On the higher level, two groups of cells are distinguished in transient configurations, namely a counter and a sphere of two waves. Initial state of the counter is determined by \( R(L) \). During generation process the states of counter are decremented until the zero state is reached. The role of counter is to control the waves. The emission of waves occurs twice, at the beginning with the speed \( v \), and at the end of counting with the speed \( 1/v \), to assure that the waves can meet. We have.

Theorem 1. Let there be given CA - a solution of the CA problem with a counter and two waves distinguished: a front one of speed \( v \) and a final one of speed \( 1/v \). Let the representation of number \( L \) is determined by \( R(L) \). Then: \( T = L/v \) and \( R(L) = L(1-v)/v \).

Proof. Is not presented here.

The lower level of organization (still above \( \lambda \)) with a set of 13 smaller components was also used in order to systematize computer aided searching of the solution. It is not explained here.

A number of 5-state solutions have been found. One of them is given in Table 1. In Fig.1 some parallelograms are shown, rows are shifted to have all counters synchronized. Following the general idea of solution the counter states are represented in transient configurations by RNS form. Thus

<table>
<thead>
<tr>
<th>( L )</th>
<th>( f )</th>
<th>( R(L) )</th>
<th>( \lambda )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

### REFERENCES

Abstract - This paper deals with computing of vertex-edge homomorphisms of deterministic labelled directed graphs in parallel. A new algorithm is presented and its time computational complexity is evaluated. A decomposition of a graph with respect to different labels is used. Number of processors required during various stages of the algorithm is assessed.

I. INTRODUCTION

Traditionally, time and memory were the only resources considered when the computational complexity of an algorithm had to be evaluated. The architecture of the computing machine was a single processor sequentially addressing the memory cells. This was a reasonable model as long as processors are much more expensive than storage space. Recent developments in VLSI technology substantially reduced the processor-to-memory cost ratio. In this technology the cost of each feature is proportional to the area it consumes on the silicon wafer, and processors and memory cells have areas of comparable size. The tradeoff between time and memory is extended to a time–hardware tradeoff, where the hardware is a combination of processors and memory. Another justification for introducing parallelism has even deeper reason than technological innovations. We notice that the remedy which shortened the length of computation time, required by the sequential algorithm, was the introduction of parallelism. Clearly, every search problem is amenable to a single time–hardware tradeoff. In addressing this problem we assume that the hardware is a shared memory computer, in which a number of processors work together synchronously and communicate with a common random access memory, in the event of read or write conflicts in this shared memory. The optimum makespan of the scheduling problem, being a function of the parallel complexity of the dag, is a parameter of the architecture.

II. PARALLEL ALGORITHM

In this what follows we will apply the following assumptions:

(i) The available number of processors is adequate for dealing with the whole width of the directed acyclic graph (dag) which represents the algorithm (thus the number of processors involved is no longer a parameter).

(ii) A communication delay $t$ between the time when some information is produced at a processor and the time it can be used by another processor is measured in elementary steps of the processors (nodes of the dag), $t$ is a parameter of the architecture.

(iii) The optimum makespan of the scheduling problem, being a function of $t$, is a fair measure of the parallel complexity of the dag, the scheduling problem is NP-complete.

We assume as a model of parallel computations the shared memory computer, in which a number of processors work together synchronously and communicate with a common random access memory, in the event of read or write conflicts in this shared memory. We assume that both conflicts are allowed, and the lowest numbered processor succeeds in the case of a write conflict.

In our approach we apply a decomposition of domain and range of directed labelled graphs with respect to different labels. There exists also a second possibility to decompose directed labelled graphs into primary graphs (see Belief [1]). Unfortunately, in the second case a decomposition of a graph is not a partition but a cover on a set of vertices, this makes load balancing more difficult (dynamic load balancing).

In algorithm description we apply the following notation:

- $A=(S_A,E_A)$, $A=(S_A,E_A)$ - a graph (state) of graph $A$ and $B$, $n_A=$ number of vertices (states) of graph $A$ and $B$,
- $A=(S_A,E_A)$, $B=(S_B,E_B)$, $n_B=$ number of different labels (inputs) of graph $A$ and $B$, $n_B=$ number of different labels (inputs) of graph $A$ and $B$, $n_B=$ number of different labels (inputs) of graph $A$ and $B$, $n_B=$ number of different labels (inputs) of graph $A$ and $B$,
- $A=(S_A,E_A)$, $B=(S_B,E_B)$ - autonomous labelled directed graphs $A$ and $B$, respectively, where $1<i<n_A$ is a transition function of graph $A$ and $B$, respectively.

SIMD parallel algorithm computing a set of generalized homomorphisms between deterministic complete labelled directed graphs:

Input: Deterministic complete labelled directed graph $A=(S_A,E_A)$ and $B=(S_B,E_B)$ with disjoint vertex sets and edge sets.

Output: Set of all generalized homomorphisms from $A$ to $B$ denoted as $GHom(A,B)$, i.e. set of state-input homomorphisms with transformations on set of vertices and on input semigroups.
Step 1: Decompose deterministic complete labelled graphs A and B into $A_1$, $A_2$, $B_1$, and $B_2$ autonomous factors, respectively; allocate these autonomous factors to $m_A$ and $m_B$ SIMD processors.

Step 2: For each autonomous factor of $A_1$ and $B_2$, respectively, compute in parallel a set of all $a_1$ connected components of $A_1$, and a set of all $b_2$ connected components of $B_2$.

Step 3: For each connected component of $A_1$ and $B_2$ compute in parallel, using at most $\min(m_A, m_B)$ processors, the following topological characteristics:

- For connected component $C_{ij}$ of $A_1$ where $1 \leq j \leq m_A$ and $1 \leq i \leq c_{ij}$:
  - lengths of cycles $d_{ij}$ for each connected component $C_{ij}$ of $A_1$,
  - level enumerations for each vertex belonging to every connected component $C_{ij}$ of autonomous factor $A_1$ and $D_j$ of autonomous factor $B_j$ (substeps a, b, c are independent and can also be performed in parallel).

- For each connected component of $A_1$ and $B_2$, compute in parallel using at most $m_A + m_B$ processors:
  - a set of generators using maximum function (a set of generators for connected component is defined as a set of all states of this component which have maximum value of vertex level enumeration, if a connected component is strongly connected then arbitrary state of this component can be treated as a generator).
  - check divisibility property of the lengths of cycles $d_{ij}$ of each connected component $D_j$ of autonomous factor $B_j$ with respect to the lengths of cycles $d_{ij}$ of each connected component $C_{ij}$ of autonomous factor $A_1$.

Step 5: For each pair of connected components $(C_{ij}, D_j)$ such that cycle length $d_{ij}$ divides $d_j$ generate nondeterministically and in parallel using at most $m_A + m_B$ processors a set of all possible mappings between sets of generators of $C_{ij}$ and set of generators of $D_j$.

Step 6: For each mapping computed in step 5 generate in parallel using at most $(m_A m_B - \Pi A, \Pi A)$ processors a binary relation of successors implied by transition functions of connected components $C_{ij}$ and $D_j$; this binary relation should be computed modulo length of cycle $d_j$ of connected component $D_j$.

Step 7: For all binary relations generated in step 6 check in parallel using at most $(m_A m_B - \Pi A, \Pi A)$ processors whether 'n' relation is a function, if such relation is a function then the corresponding automaton is strongly connected.

The dependency graph of this algorithm is presented on Fig. 1. As an example we provide a pseudocode of the parallel connected components algorithm. Implementations of other steps of the algorithm are not provided because of the lack of space.

Parallel connected components algorithm:

Step 1: Broadcast data to all processor nodes by distributing vertices and edges evenly, i.e. $n/N$ vertices and $e/N$ edges among N active processors.

Initialize each vertex to be the root of the tree which contains it.

Step 2: For each vertex $i$ and $j$ do in parallel

- If vertex $i$ and vertex $j$ are in processor
  - then process edge $(i, j)$
    - Find the root of vertex $i$.
    - Find the root of vertex $j$.
    - Merge (union) tree for $i$ and tree for $j$.
- If processor node is not a collector node
  - then pass parent array and all unprocessed edges to appropriate nodes with minimal Hamming distance.

Step 3: For collector nodes do in parallel

Combine the information in the current parent array with the parent array passed to nodes from sender node.

Repeat steps 2 and 3 for all unprocessed edges and parent arrays.

### III. REFERENCES


LU DECOMPOSITION ON A SHARED MEMORY MULTIPROCESSOR

Paul A. Farrell
Dept. of Mathematics & Computer Science
Kent State University
Kent, OH 44242, U.S.A.

Arden Ruttan
Dept. of Mathematics & Computer Science
Kent State University
Kent, OH 44242, U.S.A.

Abstract: We propose an algorithm for the parallel LU decomposition of an upper Hessenberg matrix on a shared memory multiprocessor. We consider the general case of p processors, where p is not related to the size of the matrix problem. We show that the LU decomposition of an (m + 1)-banded Hessenberg matrix can be achieved in O(3mn^2) operations, where n is the dimension of the matrix and p is the number of processors. For tridiagonal matrices this algorithm has a lower operation count than those in the literature and yields the best existing algorithm for the solution of tridiagonal systems of equations.

1. Introduction

A number of authors over the last two decades have written on parallel algorithms for solving tridiagonal systems. These articles have considered the problem of solving tridiagonal systems for the form

\[ Ax = b, \]

where A and all of the b_i are known at the start of the process. In such cases, the computations can be arranged to produce highly efficient parallel solutions to all m systems simultaneously. It should be noted, however, that there are a number of common numerical situations, for example the ADI method, where one needs to solve tridiagonal systems where A is known ab initio but the b_i's are not all known at the start of the computation but rather arise as a result of an iteration process.

2. LU Decomposition Algorithm

We shall, in fact, consider the LU decomposition of an n x n upper Hessenberg matrix, since the analysis is not significantly more difficult and the additional generality leads to insights, which produce a more efficient algorithm. Let A = (a_ij) be a banded n x n upper Hessenberg matrix with band width m + 1, i.e., a_0 = 0 only when mn {1, i - 1} \leq j \leq \max (n, m + i - 1), 1 \leq i \leq n. It suffices to consider the case where a_{m+1} \neq 0, 1 \leq i \leq n - 1, since otherwise the matrix is reducible, and we may consider the LU decomposition of the subproblems resulting from the reduction. Throughout this paper we will use the convention that any element with a nonpositive index has value zero.

As in most algorithms for shared memory multiprocessors, the object here is to partition the problem into a number of subproblems suitable for solution by tasks running on the available processors. We shall consider the general case of p processors, where p is not related to the size of the matrix problem. Clearly A has an LU factorization, A = LU, where L is a unit lower bi-diagonal n x n matrix and U = (u_ij) is a banded n x n upper-triangular, with m non zero diagonals, including the main diagonal. The special form of L allows one to readily determine L^{-1}. One finds that L^{-1} = (\tilde{L}_{ij}) is an n x n lower triangular matrix given by

\[ \tilde{L}_{ij} = \begin{cases} \frac{1}{u_{m+1}} (-\ell_j) & i \geq j \\ 0 & i < j. \end{cases} \] (1)

Thus the elements of U = L^{-1}A, satisfy 1 \leq i, j \leq n

\[ u_{ij} = \prod_{s=m+1}^{\min(n+1)} (-\ell_s)a_{sj}. \] (2)

As in the tridiagonal case, the well known substitution (cf. [2], pp. 473 - 474)

\[ y_1 = 1, \quad \ell_i = y_{i-1}/y_i \quad i = 2, 3, \ldots, n \] (3)

can be used to simplify (2). Since u_{j+1,j} = 0, for 1 \leq j \leq n - 1, (2) yields the following linear systems for the unknowns y_i.

\[ y_j = 0, \quad j \leq 0, \quad y_1 = 1 \]

\[ y_j + a_{j+1,j} = \sum_{s=m+1}^{\min(n+1)} (-1)^{s-j}a_{sj}, \quad 1 \leq j \leq n - 1. \] (4)

It is clear that (1) defines an m + 1 banded triangular linear system

\[ Ty = e. \] (5)

Thus the problem of finding an LU factorization of an upper Hessenberg matrix reduces to solving the banded-triangular system described in (5) to obtain the y_i's and then using the solution of that system to evaluate L and U. In practice, L may be determined from equation (3). To determine U, note first that the elements of the (m - 1)st diagonal, u_{i+1,i-1} satisfy u_{i+1,i-1} = a_{i+1,i-1}, for i = 1, \ldots, n + m - 1. Also, u_{1,0} = a_{1,0}, for j = 1, \ldots, m. Thus these elements do not require any calculation. The jth super-diagonal is given in terms of the (j + 1)st by

\[ \ell_i, u_{i-j,j} + u_{i-j+1} = a_{i-j+1}, i = 2, \ldots, n - j. \] (6)

Note that each element depends only on a single element of the next superdiagonal and on known values from L and A. Thus, the calculation of each super diagonal of U is perfectly parallelizable. In fact, the main diagonal may be obtained using 1 division rather than the multiplication and subtraction in (6) by

\[ u_{1,0} = \ell_1, u_{i,j} = \ell_i, u_{i,j+1} = 2, \ldots, n. \] (7)

Further, the calculation of the super-diagonals can be chained. Thus the calculation of L using (3) requires n - 1 divisions. The total computations for U is

\[ (n - 1) + 2 \sum_{j=1}^{m-2} (n - j - 1) = n(2m - 3) - (m^2 - m - 1). \] (8)

The latter term is negative for m \leq 1. Hence we get the following upper bound for the complexity of calculating U

\[ n(2m - 3). \]

Note that in the case of a tridiagonal system, m = 2. (8) reduces exactly to n - 1. Hence using p processors L and U can be calculated in the general case in

\[ \frac{2(n - 1)}{p} \] (9)

operations and in the special tridiagonal case in

\[ \frac{2(n - 1)}{p} \] (10)

operations. There remains only the solution of the triangular system Ty = e.

3. Algorithm for the Triangular System

In [3], Lakshmivarahan and Dhall present an algorithm for calculating the LU factorization of a tridiagonal matrix. Their algorithm

718
It is easily seen that after the execution of the above algorithm $D_{i,k} = (\prod_{i=1}^{(k-1)M+1} C_i)_{z_i}$. For the purpose of simplifying the complexity analysis, assume that $n = Nm + 1, p = g(2m - 1)$ where $g$ is a power of 2, and $N = gM$.

An analysis, the details of which appear in [1], yields the following time complexity estimate. For a general upper Hessenberg matrix with band width $m + 1$, the time required to calculate its $y_i$s in this fashion is

$$n \frac{6m^2 - 2m + 1 + m(2m - 1)\log\left(\frac{p}{2m - 1}\right)}{2p} + O(m). \quad (14)$$

In the tridiagonal case $m = 2$, and this reduces to

$$n \frac{21}{2} + 3\log\left(\frac{2}{3}\right). \quad (15)$$

Thus, from (9) and (14), the total cost of an LU factorization is

$$n \frac{6m^2 + 2m - 3 + (2m^2 - m)\log\left(\frac{p}{2m - 1}\right)}{2p} + O(m). \quad (16)$$
A BICONJUGATE GRADIENT TYPE ALGORITHM ON MASSIVELY PARALLEL ARCHITECTURES

Roland W. Freund AND Marlis Hochbruck
RIACS, Mail Stop Ellis Street
NASA Ames Research Center
Moffett Field, CA 94035, U.S.A.

Abstract — The biconjugate gradient (BCG) method is the "natural" generalization of the classical conjugate gradient algorithm for Hermitian positive definite matrices to general non-Hermitian linear systems. Unfortunately, the original BCG algorithm is susceptible to possible breakdowns and numerical instabilities. Recently, Freund and Nachtigal have proposed a novel BCG-type approach, the quasi-minimal residual algorithm (QMR), which overcomes the problems of BCG. Here, we present an implementation of QMR based on an s-step version of the nonsymmetric look-ahead Lanczos algorithm. The main feature of the s-step Lanczos algorithm is that, in general, all inner products, except for one, can be computed in parallel at the end of each block; this is unlike the standard Lanczos process where inner products are generated sequentially. The resulting implementation of QMR is particularly attractive on massively parallel SIMD architectures, such as the Connection Machine.

INTRODUCTION

We are concerned with the iterative solution of large sparse linear systems
\[ Ax = b, \]
where \( A \) is a nonsingular, general non-Hermitian \( N \times N \) matrix. Some of the most efficient iterative schemes for (1) are Krylov subspace methods: for any initial guess \( x_0 \in \mathbb{C}^N \), they generate approximations to \( A^{-1}b \) of the form
\[ x_n = x_0 + \sum_{i=1}^{\infty} K_i(r_0, A), \quad n = 1, 2, \ldots, \]
where \( r_0 = b - Ax_0 \) and
\[ K_n(r_0, A) = \text{span} \{ r_0, Ar_0, \ldots, A^{n-1}r_0 \} \]
is the \( n \)th Krylov subspace generated by \( r_0 \) and \( A \). For example, the generalized minimal residual algorithm (GMRES) of Saad and Schultz [8] and the biconjugate gradient algorithm (BCG) of Lanczos [6] both satisfy (2). Unfortunately, for methods such as GMRES, work and storage requirements per iteration grow linearly with \( n \), and therefore the vectors are restarted in practice, which often results in slow convergence. In contrast, for BCG, work and storage requirements per iteration are constant and low. However, BCG typically exhibits a rather irregular convergence behavior and the method can even break down.

THE QMR APPROACH

In [3], Freund and Nachtigal have proposed a BCG-type approach, the quasi-minimal residual algorithm (QMR), which overcomes the problems of BCG. The method uses an implementation developed by Freund, Gutknecht, and Nachtigal [1, 2] of the nonsymmetric Lanczos algorithm [5] with look-ahead [7] to generate basis vectors \( v_1, v_2, \ldots \) for the Krylov subspaces (3). More precisely, with
\[ V^{(n)} = \{ v_1, v_2, \ldots, v_{l(n)} \} = \{ V_1 V_2 \ldots V_l \}, \]
\[ V_k = [v_1, v_{n+k-1}, \ldots, v_{n+k-1}], \quad k = 1, \ldots, l = l(n), \]
we have
\[ K_{l(n)}(r_0, A) = \{ V^{(n)}z \mid z \in \mathbb{C}^n \} \]
for \( n = 1, 2, \ldots \).

The blocks \( V_k \) in (4) just contain the vectors corresponding to the \( k \)th look-ahead Lanczos step of length
\[ h_k = n_k+1 - n_k. \]
In the sequel, we refer to the first vectors \( v_{n+k-1} \) in each block as regular vectors, while the remaining vectors are called inner vectors. Further more, the relation
\[ A V^{(n)} = V^{(n+1)} H^{(n)} \]
holds. Here \( H^{(n)} \) is an \( (n+1) \times (n+1) \) upper Hessenberg matrix which is also block tridiagonal with \( l \) diagonal blocks of size \( h_k \times h_k \), \( k = 1, 2, \ldots, l \). In addition to the right Lanczos vectors \( v_1, v_2, \ldots \), the look-ahead Lanczos algorithm generates left Lanczos vectors \( w_1, w_2, \ldots \) such that
\[ K_n(w_1, A^T) = \text{span} \{ w_1, w_2, \ldots, w_n \} \]
for \( n = 1, 2, \ldots \), and, as in (4), we set
\[ W_k = [w_1, w_{n+k-1}, \ldots, w_{n+k-1}] \]
for \( k = 1, \ldots, l \). These vectors are just constructed such that right and left Lanczos vectors corresponding to different look-ahead steps are biorthogonal, i.e.,
\[ W_j^T V_k = \begin{cases} 0 & \text{if } j \neq k, \\ D_k & \text{if } j = k, \end{cases} \]
where \( D_k \) are all nonsingular.

By means of (5) and (6), the \( n \)th iterate (2) of any Krylov subspace method and the corresponding residual vector can be written as follows:
\[ x_n = x_0 + \sum_{i=1}^{\infty} \left( V^{(i)} z_i \right) \text{ for some } z_i \in \mathbb{C}^n, \]
\[ r_n = b - Ax_n = V^{(n+1)} \left( [1, 0, \ldots, 0] e_1 - H^{(n)} z_n \right). \]

Here \( e_1 \) denotes the first unit vector in \( \mathbb{R}^{n+1} \).

For the QMR method the parameter vector \( z_n \) in (8) is chosen such that the Euclidean norm of the coefficient vector \( z_n \) in the representation (9) is minimal, i.e., as solution of the least squares problem
\[ \min_{z \in \mathbb{C}^n} \left\| \Omega_n \left( [1, 0, \ldots, 0] e_1 - H^{(n)} z \right) \right\|_2, \]
where \( \Omega_n = \text{diag}(\|v_1\|_2, \|v_2\|_2, \ldots, \|v_{n+k-1}\|_2) \). Here, \( \Omega_n \) is chosen such that all basis vectors \( v_j / \|v_j\|_2 \), \( j = 1, \ldots, n + 1 \), in the representation (9) of \( r_0 \) have the same Euclidean length. Note that \( \Omega_n H^{(n)} \) is an upper Hessenberg matrix with full column rank. Hence (10) always has a unique solution \( z_n \) and the QMR iterate \( x_n \) is well defined by (8) and (10). Finally, we remark that \( z_n \) can be easily updated from step to step, and the resulting QMR algorithm can be implemented using only short recurrences (see [3] for details).

AN S-STEP LANCZOS ALGORITHM WITH LOOK-AHEAD

To enforce the biorthogonality conditions (7), inner products of vectors of length \( N \) need to be computed. In the implementation of the look-ahead Lanczos algorithm described in [1, 2], this is done sequentially, i.e., inner products are calculated in each iteration step. On a massively parallel machine, such as the Connection Machine, the sequential computation of these inner products represents a bottleneck.

In this section, we sketch a version of the look-ahead Lanczos algorithm which overcomes this problem and is more suited for a parallel machine. In contrast to the sequential algorithm, where look-ahead
steps of size $h_k > 1$ are performed only if necessary to avoid break
downs of the Lanczos process, the philosophy of the $s$-step Lanczos
algorithm is to construct Lanczos blocks of given size $h_s = s$, when
ever possible. This is done by first generating $s$ intermediate inner
vectors by means of simple three-term recurrences

\[ \tilde{v}_{n+1}^s = A \tilde{v}_n^s - \tilde{v}_n^s \tilde{v}_m - \gamma_n \tilde{v}_{n-1}^s, \quad (11) \]

\[ \tilde{w}_{n+1}^s = A^2 \tilde{w}_n^s - \tilde{w}_n^s \tilde{v}_m - \eta_n \tilde{w}_{n-1}^s, \quad (12) \]

with suitably chosen coefficients $\gamma_n, \eta_n$, and $\tilde{v}_m = 0$. The biortho-
gonality conditions (7) are then enforced only at the end of each block.
This has the advantage that all inner products arising in the biortho-
gonalization process for the inner vectors of a whole block can be
computed in parallel. We remark that to enforce (7) for the inner vectors in
block $I$, it is sufficient to biorthogonalize them only against the vectors
from the previous blocks $f = f(n), f + 1, \ldots$, using

\[ v_n = \tilde{e}_n - V_{f-1} D_{f-1}^T W_f^T \tilde{e}_n - \cdots - V_{f-i} D_{f-i}^T W_i \tilde{e}_n, \quad (13) \]

\[ w_n = \tilde{w}_n - W_f D_{f-1}^T V_f \tilde{w}_n - \cdots - W_i D_{i-1}^T V_i \tilde{w}_n. \quad (14) \]

Moreover, in general, only one previous block occurs in (13) and (14),
e.g., $f = l - 1$.

In [11], Kim and Chronopoulou proposed a $s$-step Lanczos algo-
rithm using a fixed block size $s$ throughout the whole process. Our
numerical tests show that such an approach is not viable. In order to
obtain a robust implementation of the $s$-step Lanczos algorithm, it is
crucial to keep the block size variable and combine the process with a
suitable look-ahead strategy.

In the following algorithm, we outline the $s$-step look-ahead Lanc-
zos method which we propose. In each block step, the algorithm tries
to build a block of size $s$. If the construction of such a block would
lead to a singular or a nearly singular matrix $D_I$ in (11), we
a new pair of $r_{n+s}$ and $w_{n+s}$ of regular vectors which have domin-
cant components in the old Krylov subspaces $K_n(s, A)$ or $K_{n+s}(s, A)$, we either build
a smaller block or, by performing sequential steps, a bigger $s$-block

**Algorithm. Sketch of $s$-step Lanczos algorithm with look-ahead**

0) Set $v_0 = r_0/||r_0||_2$ and choose $w_1 \in C^N$ with $||v_1||_2 = 1$,
Set $n_1 = 1, I = 1, \tilde{t}_0 = \tilde{v}_0 = 0$,
For $l = 1, 2, \ldots$

1) Compute $s$ - 1 intermediate inner vectors via (11) and (12) for

\[ n = n_1, \ldots, n_1 + s - 2; \]

Set $V_I = [\tilde{v}_n \cdots \tilde{v}_{n+s-1}]$, $V_I = [\tilde{w}_n \cdots \tilde{w}_{n+s-1}];$

2) Construct the symmetric matrix $W_I^T V_I$;

3) (Biorthogonalization of inner vectors.)

Determine $f$ by $n_f = \max \{ n_I | n_I \leq n_1 + s - 1 \}$.
For $n = n_1 + 1, \ldots, n_f + s - 1$, compute $v_n$ and $w_n$ via (13) and
(14): if $||v_n||_2 = 0$ or $||w_n||_2 = 0$, stop;
Set $V_I = [v_n \cdots v_{n+s-1}]$, $V_I = [w_n \cdots w_{n+s-1}];$

4) Construct the symmetric matrix $D_I = W_I^T V_I$;

5) Decide whether to construct $v_{n+s}$ and $w_{n+s}$ as regular vectors
or to reduce the block size and go to 6) or 7), respectively;

6) If it is possible to construct regular vectors $r_{n+s}$ and $w_{n+s}$ for

$g < s$:

Set $n_{g+1} = n_1 + 1$, $I = [v_n \cdots v_{n+s+1}]$, $V_I = [w_n \cdots w_{n+s+1}]$, and go to 8);
Otherwise, try to increase the block size $s$ by sequential steps:
set $s = s$;

Loop:
Set $s = s + 1, n = n_1 + s - 2$, compute $v_{n+s}$ and $w_{n+s}$ via (11) and
(12), and biorthogonalize immediately;
Determine $f$ by $n_f = \max \{ n_f | n_f \leq n_1 + s + 1 \}$ and compute
$v_{n+s}$ and $w_{n+s}$ using formulas (13) and (14); with $n$ replaced by
$n + 1$: if $||v_{n+s}||_2 = 0$ or $||w_{n+s}||_2 = 0$, stop;

7) Determine the smallest value which failed the checks and update
the upper bound $n(A)$ to this value. The block is now
enforced to close. Let its size be $s$ and set $n_{s+1} = n_1 + s$;
Set $V_I = [v_n \cdots v_{n+s+1}]$, $V_I = [w_n \cdots w_{n+s+1}];$

8) (Construct regular vectors $v_{n+s}$ and $w_{n+s}$.)

Set $n = n_{s+1}$, $\tilde{e}_n = \tilde{v}_{n+1} - \tilde{v}_n$, $A^s \tilde{w}_n = A^s \tilde{w}_n$, and compute

$\tilde{e}_n = -V_{s+i} D_{s+i}^T W_{s+i} \tilde{e}_n - V_{s+i-1} D_{s+i-1}^T W_{s+i-1} \tilde{e}_n$,

$\tilde{w}_n = -W_{s+i} D_{s+i}^T V_{s+i} \tilde{w}_n - W_{s+i-1} D_{s+i-1}^T V_{s+i-1} \tilde{w}_n$;

If $||\tilde{e}_n||_2 = 0$ or $||\tilde{w}_n||_2 = 0$, stop;
Otherwise, set $\tilde{v}_n = \tilde{v}_n/||\tilde{v}_n||_2$ and $\tilde{w}_n = \tilde{w}_n/||\tilde{w}_n||_2;

9) Construct the $l$th blocks of the block tridiagonal matrix $H^{(l-1)}$ and set $l = l + 1$.

We note that the quantity $n(A)$ in step 7) is an estimate of the
norm of the matrix $A$ which is used for our checks to guarantee that
the Lanczos vectors remain sufficiently linearly independent. A similar
concept was first introduced for the sequential look ahead Lanczos
algorithm in [11]. These checks, the criteria for the decision in step 5),
and further details of the algorithm will be presented in a forthcoming
paper. Here, we only remark that the important properties (5), (6),
and (7), which were used in the derivation of the QMR method, remain
valid also for the $s$-step Lanczos algorithm with look ahead. Also, we
note that the above algorithm can be realized with the same number
of inner products as in the classical nonsymmetric Lanczos method
without look-ahead. In particular, the $a \times a$ matrix $W_I^T V_I$ in step 2)
can be constructed by computing only $2s$ - 1 inner products, rather
than $a^2$ as the straightforward approach would suggest. Moreover,
in step 4), the matrix $D_I$ can be updated from $W_I^T V_I$ using only
already available inner products. Finally, numerical experiments with
an implementation of the resulting QMR algorithm on the CM 2 will be
reported elsewhere.

**REFERENCES**

mentation of the look-ahead Lanczos algorithm for non-Hermitian
Research Center, November 1990.

[2] R.W. Freund and N.M. Nachtigal, An implementation of the look-
ahead Lanczos algorithm for non-Hermitian matrices, Part II,
Technical Report 90.46, RIACS, NASA Ames Research Center,
November 1990.

method for non-Hermitian linear systems, Technical Report 90.51,
RIACS, NASA Ames Research Center, December 1990.

Lanczos method on parallel vector computers, Technical Report
90-38, University of Minnesota, July 1990.


[6] C. Lanczos, Solution of systems of linear equations by minimi-

algorithm for nonsymmetric matrices, Math. Comp. 44, 105 124
(1985).

[8] Y. Saad and M.H. Schultz, GMRES, a generalized minimal resid-
ual algorithm for solving nonsymmetric linear systems, SIAM J.
FAST PARALLEL SOLUTION OF SPARSE TRIANGULAR SYSTEMS

Fernando L. Alvarado
The University of Wisconsin—Madison
Madison, Wisconsin 53706, USA

Robert Schreiber
RIACS
MS T045-1, NASA Ames Research Center
Moffett Field, California 94035

Abstract: We consider parallel solution of a sparse system \( Lx = b \) with triangular matrix \( L \), which is often a performance bottleneck in parallel computation. When many systems with the same matrix are to be solved, we can improve parallel efficiency by representing the inverse of \( L \) as a product of a few sparse factors. We construct the factorization with the smallest number of factors, subject to the requirement that no new nonzero elements are created. Applications are to iterative solvers with triangular preconditioners, to structural analysis, or to power systems applications. Experimental results on the Connection Machine show the method to be highly valuable.

1. INTRODUCTION

There are two possible approaches to the parallel solution of triangular systems of equations. The usual approach is to exploit whatever parallelism is available in the usual substitution algorithm [4]. The second, which requires preprocessing, works with some representation of \( L^{-1} \).

If \( L \) is sparse, its inverse is usually much denser. Here we consider a factorization \( L^{-1} = \prod_{k=1}^{m} Q_k \) with sparse factors. Such a factorization is possible in which the factors have no more nonzeros than \( L \) [2]. The chief advantage of a factorization of \( L^{-1} \) is that all the necessary multiplications for the computation of \( Q_k x \) can be performed concurrently.

Thus, it is possible to take advantage of more parallelism in the solution of these equations.

We review the use of partitioned inverses of \( L \). Any triangular matrix \( L \) can be expressed as a product of elementary matrices: \( L = L_1 L_2 \cdots L_{n-1} \). The factor \( L_j \) is lower triangular and nonzeros lie below the diagonal only in column \( j \), i.e. it is elementary lower triangular. Regrouping, we may write

\[
L = \prod_{k=1}^{m} P_k
\]

where \( P_k = L_{e_k-1+1} L_{e_k-1+2} \cdots L_{e_k} \) and

\[
0 = e_0 < e_1 < \cdots < e_m = n-1.
\]

Here \( (e_k)_{k=0}^m \) is a monotonically increasing integer sequence. The factor \( P_k \) is lower triangular and is zero below its diagonal in all columns except column \( e_k-1+1 \) through \( e_k \), where it is identical to \( L \).

The solution of the partitioned problem proceeds as follows. From (1) it follows that

\[
x = L^{-1} b = \prod_{k=1}^{m} P_k^{-1} b.
\]

In computing the matrix-vector products, we may exploit parallelism fully, using as many processors as there are nonzeros in \( P_k \) and summing the results in logarithmic time.

A. Problems Addressed

We say that the matrix \( X \) is invertible in place if \( x_{ij} \neq 0 \) \( \Leftrightarrow (X^{-1})_{ij} \neq 0 \). The elementary lower triangular matrices are invertible in place. There therefore always is at least one partition (1) of \( L \) with factors that invert in place.

\[\*\]This work was supported in part under NSF Contracts ECS-8823654 and ECS-8907391.

\[\*\*\]This work was supported by the NASA Systems Division under Cooperative Agreement-NCC 9-397 between NASA and the University Space Research Association (USRA).

Figure 1: Best no-fill partition for a graph without reordering. Five factors are required.

Figure 2: Best no-fill partition for a reordered graph. Only three factors are required.

Definition 1 A partition (1) in which the factors \( P_k \) are invertible in place is called a no-fill partition. A no-fill partition of \( L \) with the smallest possible number of factors, is a best no-fill partition.

Let \( G(L) \) be the digraph with vertices \( V = \{1,2,\ldots,n\} \) and directed edges \( E = E(L) = \{ (i,j) \mid i > j, L_{ij} \neq 0 \} \). \( G(L) \) is an acyclic digraph, or DAG. Consider the matrix \( L \) with graph \( G(L) \) illustrated in Figure 1. \( L \) has a best no-fill partition:

\[
\]

This partition has six factors. It is possible to symmetrically permute the rows and columns of \( L \) such that \( L \) remains a lower triangular and \( G(L) \) is as illustrated in Figure 2. A best no-fill partition of this reordered \( L \) is

\[
L = (L_1 L_2)(L_3 L_4)(L_5 L_6 L_7),
\]

which has only three factors.

722
Definition 2 A best reordered partition of $L$ is a symmetric permutation of the rows and columns of $L$ such that the permuted matrix is triangular and has a best no-fill partition with the fewest possible factors.

The aims of this work are as follows. We shall develop a theory of efficient algorithms for computing best no-fill (Section II) and best reordered (Section III) partitions. Second, we shall determine how useful these ideas are in practice by means of some experiments (Section IV).

II. BEST NO-FILL PARTITIONS

The transitive closure of a digraph $G = (V, E)$ is the graph $G^t = (V, E')$ where $E' = \{(i, j) \mid \text{there is a } j \to i \text{ path in } G\}$. The digraph $G^t = (V, E')$ is transitively closed if it is equal to its own transitive closure.

We shall state our results and refer the reader to the full paper [1] for proofs.

**Lemma 1** If $L$ is a nonsingular lower triangular matrix, then $G(L^{-1})$ is the transitive closure of $G(L)$.

Let $G = (V, E)$ be a digraph associated with a triangular matrix $L$. Given a subset $S$ of $V$, define the column subgraph $G_S = (V, E_S)$ (where $E_S = \{(i, j) \in E \mid j \in S\}$) as the graph of the lower triangular matrix obtained by zeroing all columns of $L$ not in $S$.

**Theorem 2** Let a partition $(2)$ and corresponding factorization $(1)$ be given. The factors $P_k$ are invertible in place if each column subgraph $G(P_k) = G_{S_k1 \ldots S_k}$ is transitively closed.

**Proof:** By Lemma 1 $(P_k^{-1})_{ij} \neq 0$ iff there is a $j \to i$ path in $G(P_k)$. The following are therefore equivalent:

- $P_k$ is invertible in place;
- $(P_k^{-1})_{ij} \neq 0 \Rightarrow (P_k)_{ji} \neq 0$;
- for every $j \to i$ path, $(P_k)_{ji} \neq 0$;
- $G(P_k)$ is transitively closed.

The following algorithm was proposed by Alvarado, Yu and Betancourt [2];

**Algorithm P1:**

**Input:** $L = L_1 E_2 \cdots L_{n-1}$

**Output:** A best no-fill partition of $L$.

```plaintext
i ← 1;
while (i < n - 1) do
    k ← i + 1;
    while (i < n - 1) do
        Let $r$ be the largest integer greater than $i$ such that $L_i \cdots L_r$
        is invertible in place;
        $P_k ← L_i \cdots L_r$;
        $k ← k + 1$;
    i ← r + 1;
enddo;
enddo
```

We have shown [1] Algorithm P1 determines a best no-fill partition. (There may be others. Best no-fill partitions are not unique.)

III. BEST REORDERED PARTITIONS

This section describes a straightforward "greedy" algorithm for finding best reordered partitions.

**Definition 2** For $(i, j) \in E$ we say that $j$ is a predecessor of $i$ and $i$ is a successor of $j$. Let $G = (V, E')$ be a DAG. We define level$(i), i \in V$ to be the length of the longest path in $G$ ending at $i$.

The algorithm works by finding a partition $V = \bigcup_{k=1}^m S_k$ for which the column subgraphs $G_{S_k}$ are transitively closed. Moreover, $S_1$ is a source node in the quotient graph, i.e. there are no edges directed into $S_1$. If $S_1$ and its out edges are removed, then $S_2$ is a source node, etc. We shall call the subsets $S_k$ in this partition factors in analogy with the corresponding factors $P_k$ of $L$.

Algorithm RP1 (Re-order, Permute 1):

**Input:** A directed, cyclic digraph $G(L)$.

**Output:** A permutation $\pi : V \rightarrow \{1, \ldots, n\}$ and a partition of $L$.

```plaintext
Compute level$(v)$ for all $v \in V$;
max-level ← max$_{v \in V}$level$(v)$;
i ← 1;
k ← 1;
e_0 ← 0;
while $i < n$ do
    $S_k ← e_0$;
e_0 ← $e_k$ + 1;
enddo;
```

**Theorem 3** Procedure RP1 finds a best reordered partition of of $L$.

The complexity of Algorithm RP1 can be large. Consider a dense lower triangular matrix of order $n$. RP1 takes $O(n^3)$ time in this case, since the cost of checking whether all successors of vertex $j$ are also successors of its predecessors is $O(j(n - j))$. In [2] we refined Algorithm RP1, producing an improved algorithm for which we can prove an $O(n^2)$ complexity bound. Furthermore, A. Pothen has developed a method with an $O(n^2)$ complexity bound for the case where the undirected graph $G(L + L^T)$ is chordal (this happens when $L$ is a Cholesky factor) [9].

While our work is related to the use of clique and clique tree representations of sparse matrix factors [7], the partitioning of this paper is not the same as the partitioning into simplicial cliques or supernodes that has appeared previously. Indeed, all members of a simplicial clique of $G(L)$ are included in the same factor by Algorithm RP1, but several simplicial cliques may belong to the same factor; see [1].

IV. EXAMPLES

This section illustrates the performance of the proposed algorithms. Several examples compare P1 and RP1 with respect to the number of factors in the partitions they find. We also examine the effect of the initial ordering of rows and columns of a matrix $A$ on the number of factors in a best reordered partition of its Cholesky or incomplete Cholesky factor $L$. An experiment shows that on the Connection Machine, the solution process (3) can be faster than substitution by orders of magnitude.

First, we compare algorithms P1 and RP1. Table 1 uses five power system matrices ranging in size from 118 to 993. Table 2 gives results for matrices arising from five-point finite difference discretizations. In each case, the original coefficient matrix is first ordered and its Cholesky factor $L$ is found. We need to distinguish this first fill-reducing ordering of $A$ from the reordering of $L$ found by RP1. We call the ordering of $A$ the primary ordering. Three primary ordering procedures are used: the minimum degree algorithm [8], the multiple minimum degree (MMD) algorithm [5], and the minimum level, minimum degree (MLMD) algorithm [3].

For each matrix and primary ordering algorithm, two partitioning methods are compared: Algorithm P1, which simply partitions $L$ optimally without reordering it, and Algorithm RP1 which reorders the matrix and generates an optimal partition. In most cases, Algorithm

723
The computations used 8192 processors on the Connection Machine the NAS Systems Division, NASA Ames. From these results, as well as those above, we see that unless \( L \) has a fairly rich structure there is no great advantage to the use of the partitioned method. The use of an MLMD primary ordering improves both substitution and partitioned methods. However, with the introduction of the additional fill in the exact factor \( L_3 \) (compared with \( L_2 \)), the number of levels in \( G(L) \) increases sharply (as does the time for substitution) while the number of factors in the best reordered partition drops dramatically. The difference in the solution time, even for this problem of modest size, is about a factor of twenty. Thus, we conclude that the method can be quite useful in highly parallel machines when the matrix \( L \) has a rich enough structure, as happens when it is an exact triangular factor.

### V. CONCLUSIONS

An algorithm for best reordered partitioning of lower triangular matrices has been presented and proven to be optimal.

Experiments have shown the method to be very valuable in practice highly parallel machines.

For a lower triangular factor of a sparse matrix \( A \), the number of partitions attainable is strongly influenced by both the ordering of rows and columns of \( A \) and the method of computing \( L \).

### References


### Table 1: Effect of primary ordering on number of factors in best no fill partitions (P1) and best reordered partitions (RP1) for power system matrices.

<table>
<thead>
<tr>
<th>Size</th>
<th>Min. Degree</th>
<th>MMD</th>
<th>MLMD</th>
</tr>
</thead>
<tbody>
<tr>
<td>118</td>
<td>53</td>
<td>14</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>352</td>
<td>132</td>
<td>21</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td>707</td>
<td>213</td>
<td>26</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>18</td>
<td>11</td>
</tr>
<tr>
<td>1084</td>
<td>309</td>
<td>26</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>24</td>
<td>14</td>
</tr>
<tr>
<td>1993</td>
<td>563</td>
<td>35</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>25</td>
<td>15</td>
</tr>
</tbody>
</table>

### Table 2: Effect of primary ordering on number of factors in best no fill partitions (P1) and best reordered partitions (RP1) for 5-point difference operators on grid graphs.

<table>
<thead>
<tr>
<th>Grid Size</th>
<th>Min. Degree</th>
<th>MMD</th>
<th>MLMD</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 by 5</td>
<td>10</td>
<td>17</td>
<td>6</td>
</tr>
<tr>
<td>10 by 10</td>
<td>20</td>
<td>12</td>
<td>15</td>
</tr>
<tr>
<td>15 by 15</td>
<td>20</td>
<td>12</td>
<td>14</td>
</tr>
</tbody>
</table>

In Table 3 we give the size of these factors, the number of levels, which is proportional to the time required for our parallel substitution algorithm, and the number of partitions, which is in practice proportional to the time required by the partitioned solution algorithm (3).

### Table 3: Comparison of CM-2 execution times for substitution and partitioned solution.

<table>
<thead>
<tr>
<th>Matrix</th>
<th>Ordering</th>
<th>Factor-</th>
<th>nonzeros</th>
<th>Levels in ( G(L) )</th>
<th>Substitution Time</th>
<th>Factors</th>
<th>Partitioned soln. time</th>
</tr>
</thead>
<tbody>
<tr>
<td>( L_1 )</td>
<td>RCM</td>
<td>ILU</td>
<td>23,526</td>
<td>823</td>
<td>16.17 secs</td>
<td>816</td>
<td>15.73 secs</td>
</tr>
<tr>
<td>( L_2 )</td>
<td>MLMD</td>
<td>ILU</td>
<td>26,793</td>
<td>78</td>
<td>2.26 secs</td>
<td>66</td>
<td>1.84 secs</td>
</tr>
<tr>
<td>( L_3 )</td>
<td>MLMD</td>
<td>exact</td>
<td>118,504</td>
<td>311</td>
<td>28.89 secs</td>
<td>16</td>
<td>1.51 secs</td>
</tr>
</tbody>
</table>
Abstract: The prototype supercomputer and some leading architectures including massively parallel computers are discussed. A performance formula explains the gap between theoretical peak and real performance. The price/performance relation for the discussed computers is presented.

1. THE PROTOTYPE VECTOR COMPUTER

In Fig. 1 the 'prototype' vector computer is depicted. The technology is characterized by the cycle time in nsec. The floatingpoint units are pipelines, thus have a startup time and need long vectors for efficient use. Chaining means coupling and overlapped operation of different pipelines, e.g. load, multiply, add, multiply.

Fig. 1 'Prototype' vector computer.

store. There may be internal parallelism by multi-track pipelines, e.g. 4-track pipelines deliver 4 results per cycle.

In this paper we denote by 'word' 64 bits = 8 bytes and by 'pipe group' an addition and multiplication pipe. Critical points of vector computers are: memory bandwidth in words per cycle and pipe group; memory size in Mwords (million words); size and bandwidth of extended memory (the two sizes limit the need 2 loads and 1 store per cycle and pipe group. Critical points of vector computers are: EM. Soon expected: Fujitsu VP2600 (Siemens S600), cycle time 3.2 nsec, theoretical peak performance 5.0 GFLOPS, monoprocessor, but internally 4-2 track pipe groups, 256 Mwords MM: 512 banks, 14 cycles bbt, bandwidth: one word per cycle and pipe group, 1 Gword EM: bandwidth 2 GB/sec. Model VP2600/20 with two scalar processors, model VP240/40 with two vector (2.2 track pipe groups) and 4 scalar processors.

NEC SX-3: Model 44, cycle time 2.9 nsec, theoretical peak performance 22 GFLOPS, 4 processors (MIMD-shared memory), 8-track pipe group, 256 Mwords MM: 1024 banks, 7 cycles bbt, bandwidth: 0.5 words per cycle and pipe group globally (between memory and memory access unit), but 1 load and 0.5 store per cycle and pipe group locally (between memory access unit and processor), 2 Gwords EM: bandwidth 2.75 GB/sec. One or two control processors for operating system, with separate memory. Remark. Only the two processor Model 24 can be recommended (11 GFLOPS) if at least 1 word per cycle and pipe group global memory bandwidth is requested.

IBM ES-9000/720 with 6 VEs (Vector Facilities = integrated vector processors), corresponds to former 3090/600J, basically general purpose computer, cycle time 14.5 nsec, theoretical peak performance 82 MFLOPS, 6 processors (MIMD shared memory), 64 Mwords MM: transparent (no banking), bandwidth: one word per cycle and pipe group, 512 Mwords EM: bandwidth 2 * 138 MB/sec. Cache of 256 KB, danger of cache stalling for long vectors. Announced: ES 9900/9000, cycle time 9.5 nsec, 2536 MFLOPS, 6 processors, improved cache, two-track pipes.

CONVEX C240. Minisupervisor, cycle time 40 nsec, theoretical peak performance 200 MFLOPS, 4 processors (MIMD shared memory), 256 Mwords MM: 128 banks, 8 cycles bbt, bandwidth: one word per cycle and pipe group, no EM. Soon expected: C300, cycle time 16 nsec, 8 processors, 1 GFLOPS theoretical peak performance, 312 Mwords MM.

IBM RISC SYSTEM 6000. Model 308. Super-Workstation, cycle time 24.4 nsec, theoretical peak performance 82 MFLOPS (can 'simulate' vector operations), 64 Mwords MM, transparent, bandwidth: one word per cycle.

2. ARITHMETIC OPERATIONS AND MEMORY BANDWIDTH

Dyadic operations like

\[ c_i = a_i + b_i \]

need 2 loads and 1 store per cycle and pipe group. Triadic operations allow parallel operation of the addition and multiplication pipeline and deliver two results per cycle and pipe group, called superscalar speed. The vector (or full) triad

\[ d_i = a_i + b_i * c_i \]

needs 3 loads and 1 store per cycle and pipe group. This is the most important operation and thus our key operation. More special is the linked triad with one scalar operand

\[ c_i = a_i + s * b_i \]

(3)

with two loads and one store and still more special is the (repeated) contracting linked triad

\[ b_i = a_i + s * a_i \]

(4)

that needs only one load if b is fixed in a vector register. This is the basic operation of matrix multiplication. A vector computer with a memory bandwidth of one word per cycle and pipe group fits only to (4) and delivers only 1/4 of the peak performance for (2).

Because of the different number of memory references one should not count for vector computers merely additions and multiplications but count superscalar triads like e.g. linked triads.

3. SOME VECTOR COMPUTER ARCHITECTURES

In [1] detailed discussions of the most relevant supercomputers, including kernel program measurements, are presented. Present supercomputers are the continuation of these architectures. In the oral presentation overviews like Fig. 1 are presented for the different supercomputers. Here we present only their main characteristics for the maximal configurations. We discuss 3 real supercomputers, an integrated vector processor, a minisupercruber and a super- workstation.

CRAY Y-MP/S2/S6, cycle time 6 nsec, theoretical peak performance 2.67 GFLOPS, 8 processors (MIMD shared memory computer), 256 Mwords MM (main memory). 256 banks, 5 cycles bbt (bus busy time), bandwidth: 2 load and 1 store per cycle and pipe group, 2 Gwords EM (extended memory): bandwidth 2 * 1.3 GB/sec.

NEC SX-3: Model 44, cycle time 2.9 nsec, theoretical peak performance 22 GFLOPS, 4 processors (MIMD-shared memory), 8-track pipe group, 256 Mwords MM: 1024 banks, 7 cycles bbt, bandwidth: 0.5 words per cycle and pipe group globally (between memory and memory access unit), but 1 load and 0.5 store per cycle and pipe group locally (between memory access unit and processor), 2 Gwords EM: bandwidth 2.75 GB/sec. One or two control processors for operating system, with separate memory. Remark. Only the two processor Model 24 can be recommended (11 GFLOPS) if at least 1 word per cycle and pipe group global memory bandwidth is requested.

IBM ES-9000/720 with 6 VEs (Vector Facilities = integrated vector processors), corresponds to former 3090/600J, basically general purpose computer, cycle time 14.5 nsec, theoretical peak performance 82 MFLOPS, 6 processors (MIMD shared memory), 64 Mwords MM: transparent (no banking), bandwidth: one word per cycle and pipe group, 512 Mwords EM: bandwidth 2 * 138 MB/sec. Cache of 256 KB, danger of cache stalling for long vectors. Announced: ES 9900/9000, cycle time 9.5 nsec, 2536 MFLOPS, 6 processors, improved cache, two-track pipes.

CONVEX C240. Minisupervisor, cycle time 40 nsec, theoretical peak performance 200 MFLOPS, 4 processors (MIMD shared memory), 256 Mwords MM: 128 banks, 8 cycles bbt, bandwidth: one word per cycle and pipe group, no EM. Soon expected: C300, cycle time 16 nsec, 8 processors, 1 GFLOPS theoretical peak performance, 312 Mwords MM.

IBM RISC SYSTEM 6000. Model 308. Super-Workstation, cycle time 24.4 nsec, theoretical peak performance 82 MFLOPS (can 'simulate' vector operations), 64 Mwords MM, transparent, bandwidth: one word per cycle.

4. PARALLEL COMPUTERS

We denote by 'parallel computer' an architecture that can be extended to 'many' processors. With parallel computers the real problems are shifted from the hardware level to the software level, i.e. to the user. We discuss only MIMD-type parallel computers. There are three basic types, see Fig. 2.

The '1. - " of shared memory computers is the memory bottleneck that limits the number of processors, and the memory contention. The user has to distribute only the processing.

The problem of the message passing system is the distribution of the data to the local memories, and the reorganization of the data. The processors of a dedicated (sub)system cannot be used by other jobs, thus we have a GFLOPS PC. The dream of the message passing community is the virtual shared memory, but this is just contrary to the communication efficiency. Hybrid systems are the worst of all worlds because they combine the disadvantages of both inherent systems.

An essential drawback of parallel computers is that efficient programs must be tailored to the special architecture. In the following we discuss briefly two
representative hypercube architectures, in the total presentation detailed overview
graphs will be presented.

NCUBE2: max. 13-dimensional hypercube, 8 - 8192 scalar 64-bit processors,
max. 20 GFLOPS for 64 bit. The basic processor is a 1.2 micron CMOS custom
processor with 0.5 M transistors, 20 MHz, 50 nsec cycle time, delivering 2.4
MFLOPS (the pipeline formula is not applicable), with 14 bidirectional DMA
channels with 2.2 MB/sec, each (one is for i/o). One node has 1 or 4 or 16 MB
local memory. Subcubes can be allocated to different users (space sharing,
PCS). The job management, the compilers and the tools are running on the
SUN workstation that serves as host computer.

INTEL iPSC/860: max. 7-dimensional hypercube, 8 - 128 i860 vector processors,
max. 51 GFLOPS for 64 bit. The basic i860 processor is a 0.5 micron CMOS
custom processor with 1 M transistors, 40 MHz, 25 nsec cycle time. The
pipelined floating point units deliver 40 MFLOPS 'supervisor speed' because the
multiply unit needs 2 cycles for one result. Unfortunately presently the software
is far behind the hardware, and Fortran compilers are unable to reach
this performance. There is a one-word (64 bits) per two cycle memory
bottleneck between i860 chip and memory. The node has
8 or 16 MB local memory and 8 bidirectional communication links with 2.8 MB/sec, each (one is for i/o). Subcubes can be allocated to different users (space sharing: 'PCS'). The job
management, the compilers and the tools are running on an INTEL 80386 PC
that serves as host computer.

5. PERFORMANCE FORMULA

In this section we want to explain why there is such a large gap between
theoretical peak performance and real performance for present supercomputer
architectures. If we solve 1000 \* vector triad, where we measure the cycle time \( t \) in nsec, we get the (vector) speed of a single pipeline in MFLOPS, times two yields the
supervisor speed, times the number \( P \) of total pipe groups in the system yields
the theoretical peak performance. But unfortunately the latter is reduced by
several reduction factors \( f_t \) that we want to discuss below. Thus our formula for
the real performance of a pipelined supercomputer is

\[
\text{r}_{\text{real}} = \frac{1000}{t \times \text{[MFLOPS]}} \times 2 \times P \times f_t \times f_2 \times \ldots \times f_n \quad \text{(5)}
\]

where we measure the cycle time \( t \) of the processor and \( f_t \) is the global utilization factor.

Repeatedly internally lost cycles (e.g. section loop organization): for \( m \) cycles
we have in the mean \( m \times d \) lost cycles, thus

\[
f_t = \frac{1}{1+4d} \quad \text{(6)}
\]

e.g. \( d = 0.05, f_t = 0.95 \).

Memory bottleneck for \( \text{vector triad} \) (2). only the \( \text{vector triad} \) has full flexibility of
multiplication and addition as 'summed' in the peak performance, all other
units are 'exceptions'. Therefore we select as \( f_2 \) the operation the \( \text{vector triad} \) that
needs \( d \) memory references per cycle and pipe group. If we have instead of \( d \)
only \( m \), we get

\[
f_2 = \frac{m}{4} \quad \text{(7)}
\]

e.g. \( m = 1, f_2 = 0.25 \).

Finite vector loop \( n \) (startup cycles lost once), If we denote by \( n_{1/2} \) Hockney's
half performance length, see [1,2], we get

\[
f_2 = \frac{n}{n + n_{1/2}} \quad \text{(8)}
\]

e.g. \( n = 1000, n_{1/2} = 1000, f_2 = 0.91 \).

Scalar code, part of operations is vectorizable, part \( (1-x) \) is scalar, \( w \) is the
ratio vector/scalar speed for infinitely long vectors. Then we get

\[
f_2 = \frac{1}{1-(1-x)w} \quad \text{(9)}
\]

This is Amdahl's law for vectorization, see [1,2], e.g. \( w = 10, v = 0.05, f_1 = 0.96 \).
Up to now we have discussed monoprocessor vector computers. For
the contemporary values we get \( f_t \) \* \( f_2 \) \* \( f_1 \) \* \( f_1 \) \* \( f_1 \) \* \( f_1 \). Also, our supercomputer
would deliver only 15 % of its theoretical peak performance for the \( \text{vector triad} \).

In the following we discuss parallel computers with \( p \) processors.

Sequential (non-parallelizable) code, part \( q \) of operations is parallelizable, part
(1-\( q \)) is sequential (fine grain parallelism). Then we get

\[
f_t = \frac{1}{(1-q)p+q} \quad \text{(10)}
\]

This is Amdahl's law for parallelization, compare to (9), e.g. \( p = 64, q = 0.95, f_t = 0.04 \).

Shared memory MIMD parallel computer, memory contention. If \( a \) denotes the
time of available virtual address (42) memory banks and \( d \) denotes the
part of operations with contiguous, \( (1-c) \) with random elements (indirect
addressing), then a model yields

\[
f_{a} = \frac{1}{1+2(1-c)a} \quad \text{(11)}
\]

e.g. \( a = 2, c = 0.95, f_{a} = 0.95 \).

Message passing (distributed memory) MIMD parallel computer, waiting for
communication. For \( m \) useful cycles in the mean \( m \times b \) additional cycles are
needed for non-overlapping communication. Then we get

\[
f_{a,m} = \frac{1}{1+b} \quad \text{(12)}
\]

e.g. \( b = 0.05, f_{a,m} = 0.95 \).

If we calculate up to this point \( f_t \) \* \( f_2 \) \* \( f_1 \) \* \( f_1 \) \* \( f_1 \) = 0.034, we see that we get with
those 'reasonable' assumptions finally not more than 3.4 % of the peak
performance, i.e. we have lost 96.6 % of our supercomputer!

Local balancing (course grain parallelism). We assume that an \( m \) has reserved
\( p \) processors (e.g. has a subcube \( t \) is the time that processor \( i \) is active. Then we get
his 'personal' utilization factor

\[
f_t = \frac{\sum_{i} f_t}{p \times \max f_t} \quad \text{(13)}
\]

e.g. \( f_t = 0.9 \).

Long-range continuous usage: an idling computer produces no GFLOPS. Thus
the global utilization factor

\[
f_g = \frac{\text{hours of usage per year}}{8760} \quad \text{(14)}
\]

is of decisive importance. For a workstation that is used (as number cruncher)
for 8 h on 5 days a week \( f_g = 0.24 \).

6. PRICE/PERFORMANCE RELATIONS

The price/performance relation has two components. The \textit{price} is determined
by the selected configuration that is strongly determined by the size of the
memory. We select a configuration for the solution of \textit{large} problems, else a
supercomputer is not an appropriate tool. Therefore we select (as far as possible)
1 GB MM (main memory), 1 GB EM (extended memory) and 100 GB
disks. \( P \) denotes the purchase price, given in MDM (million deutschmark), \( M+L \)
denotes maintenance and licence costs, given in MDM/a (per annum = year).
All prices are commercial list prices (no university discount), without VAT, valid
January 1991. The software for a FORTRAN environment with operating
system, compiler and tools is included.

The performance is given by the peak performance and the reduction factors.
We determine the performance for the \textit{vector triad}. We note only the reduction
factors \( \times 1 \) that are applied.

We attribute some personnel to the computer. SE denotes system engineer with
78 KDM/a, OP denotes operator with 52 KDM/a. We do not consider cost for
housing, electricity, climatization.

The \textit{price/performance relation} is given in MDM/a per GFLOPS, i.e. the cost
to be paid per year to have 1 GFLOPS sustained \textit{vector triad}. For this purpose
the purchase price is distributed onto \textit{four years}, i.e. we assume a four year life-
cycle of our supercomputer. In the following we give the data for the different
computers.

\begin{itemize}
\item \textit{CRAY Y-MP}38/1328: 1 GB MM, 1 GB EM, 100 GB disks, \( P = 48.05 \) MDM (31
\textit{MS}), \( M+L = 2.16 \) MDM/a, \( 2 \) SE, \( 3 \) OP, peak performance 2.67 GFLOPS, \( f_g = 0.9 \) \textit{estimated}.
\item \textit{Fujitsu Siemens VP200/10}: 1 GB MM, 1 GB EM, 100 GB disks, \( P = 28.65 
\textit{MDM}, \( M+L = 1.48 \) MDM/a, \( 2 \) SE, \( 3 \) OP, peak performance 5.8 GFLOPS, \( f_g = 0.94 \)
\end{itemize}
Nec SX-3: Model 24: 1 GB MM, 1 GB EM, 100 GB disks, P = 30.48 MDPM, M+L = 1.37 MDPM/a, 2 SE, 3 OP, peak performance 11.03 GFLOPS, f_s = 1/4, f_{cache, stubble} = 0.95 (estimated).

IBM ES9000/720 (6000/6000): CPU: 0.5 GB MM (max), 1.5 GB EM, 100 GB disks, P = 38.24 MDPM, M+L = 1.829 MDPM/a, 2 SE, 3 OP, peak performance 0.829 GFLOPS, f_s = 1/4, f_{cache, stubble} = 0.8 (estimated).

Convex C240: 1 GB MM, 8 GB disks, P = 3.9 MDPM, M+L = 0.39 MDPM/a, 1 SE, peak performance 0.2 GFLOPS, f_s = 1/4, f_{cache, stubble} = 0.9 (estimated). (Remark: 1 GB MM is unusually large for a CONVEX and brings the price up, but for the solution of large problems we need a large MM.)

IBM RISC System/6000: Model 550 Workstation: 0.5 GB MM (max), 2.5 GB disks, P = 953 KDM, M+L = 26.5KDM/a, 0.1 SE, peak performance 0.081 GFLOPS, f_s = 1/4, f_T = 0.24 or 0.75. (Remark: The maximal MM of 0.5 GB excludes this workstation from the solution of very large problems. The price for the memory brings the price for the workstation up.)

NCUBE: Model 10: 512 processors à 4 MB = 2 GB MM, 50 GB disks, P = 7.0 MDPM, M+L = 0.70 MDPM/a, 2 SE, peak performance 1.23 GFLOPS, f_s = 1 (scalar), f_{cache, stubble} = 0.95 (f = 0.9999), f_T = 0.1 or 0.24 or 0.75. (Remark: The 2.4 MFLOPS per processor is based on parallel scalar execution of addition and multiplication and are met for the vector triad by the 80 MB/sec between CPU and its local memory.)

INTEL IPSC/660: Model 128: 128 processors à 16 MB = 2 GB MM, 50 GB disks, P = 9.8 MDPM, M+L = 0.98 MDPM/a, 2 SE, peak performance 5.1 GFLOPS, f_s = 1/4, f_{cache, stubble} = 0.5 (f = 0.9999), f_T = 0.1 or 0.24 or 0.75. (Remark: The value f_s = 1/4 results from the fact that a load for 64 bits needs two cycles, but an immediately following store needs 3 cycles, thus 9 - 4.5 * 2 = 0.5 cycles are needed for the vector triad and 40/4.5 = 8.9 MFLOPS per 1860 chip result. An assembler program should come close to this value, a vectorizing compiler in a test has obtained 7 MFLOPS according to INTEL. But the software that has been delivered up to now is far behind the hardware possibilities.)

Everybody is free to choose his own parameters in this play. In Fig. 3 the price/performance relation for our parameters is depicted. The results speak for themselves. The general purpose computer has its own merits, but it is an expensive number cruncher. Nevertheless the purchase of VFs pays if the ES9000 is used in scientific computations. The mini-supercomputer is relatively expensive in spite of the 'cheap' technology because of the relatively large memory. For the workstation and the parallel computers the utilization factor is decisive for the price/performance relation. These computers have (not yet) a bulk environment that allows the continuous usage of the computer like for a conventional computer.

We have presented list-prices. A discount may change the relations correspondingly. Note that the software results in additional losses that may increase the price/performance relation considerably. But finally an efficient use of any type of supercomputer is possible only with data structures that are tailored to its architecture. If the data structure does not allow sufficient vectorization and/or parallelization, Amdahl's law (9) and/or (10) will destroy any efficiency.

7. CONCLUDING REMARKS

The cycle time for the ECL-technology of the large supercomputers will continue to decrease from now 2.9 nsec (NEC SX-3) to perhaps 2 nsec (CRAY-3?) and eventually 1 nsec for 1993++. The CMOS-Technology of presently 40 MHz/25 nsec (6800 chip) will evolve to 50 - 80 MHz/20-12.5 nsec (12.5 nsec was the cycle time of the CRAY-1, INTEL expects for the year 2000 200 MHz/5 nsec. Thus the usual factor of 10 in speed between ECL and CMOS may be maintained.

The engineers want (sustained) performances of 100 GFLOPS, then 1000 GFLOPS = 1 TFLOPS. This can be obtained only by parallelism. But there are two main problems:

Problem 1. Can we pay the memories that are needed to store the operands in order to use TFLOPS? The answer is: We have to wait until we can afford these memories. Increased parallelism does not solve this problem.

Problem 2. How can we organize user-friendly parallelism on the hardware level? The answer is presented in [3].

REFERENCES

Supercomputers: Numerical Libraries

Nikolaus Geers
University of Karlsruhe, Computing Center
Postfach 6980
D-7500 Karlsruhe

Abstract

Different numerical libraries from hardware manufacturers and software houses are available for supercomputers. Contents and performance of these libraries on CRAY 2, CRAY Y-MP, Fujitsu/Siemens VP-series and IBM 3090 VF is discussed and a proposal for future library development is given.

1 Introduction

Numerical subprogram libraries are important tools in developing scientific software. With the increasing use of supercomputers, in most cases pipelined vector computers, there is an increasing need for reliable and efficient collections of numerical subprograms. This frees the application programmer from re-encoding numerical algorithms and increases the reliability and maintainability of application programs.

This paper will concentrate on discussion of efficiency of libraries on supercomputers but will not include a general evaluation of different libraries since this must comprise a detailed discussion of the numerical algorithms, their robustness and their implementation which is beyond the scope of this paper.

In section 2 a brief review of different libraries for supercomputers will be given. Since most work on adapting libraries to supercomputer architectures has been done in the area of linear algebra this is discussed in greater detail in section 3. In the concluding remarks some items missing in today's subroutine libraries are listed and a proposal for future library developments on supercomputers is given.

2 Numerical Libraries for Supercomputers

Two groups of numerical subroutine libraries for supercomputers must be distinguished:

- Libraries developed by hardware manufacturers and being a part of the software environment on a given computer system,
- Libraries developed by software vendors (e.g. IMSL and NAG) which are in general available on a wide range of different computer systems.

2.1 Libraries from Computer Manufacturers

Cray: SCILIB The Scientific Library (SCILIB), release 5.0, contains programs from three different areas: linear algebra, Fast Fourier Transforms (FFT) and searching and sorting. The linear algebra chapter comprises all level 1, level 2 and level 3 BLAS as described in [6, 1, 2] except those level 2 BLAS for packed matrices. Also optimized versions of LINPACK and EISPACK are included in SCILIB. There are routines for single and multiple FFT's, the routines doing multiple transforms are of mixed radix type allowing radices 2, 3, 5 and 7. A random number generator is included in the runtime library of the compiler.

IBM: ESSL The Engineering and Scientific Subroutine Library (ESSL) from IBM is available in a vector as well as in a scalar version running on any IBM mainframe. In addition to the chapters on linear algebra, FFT, sorting and searching some other routines for interpolation, numerical quadrature, random number generation and parallel processing are included in this library. Only a subset of level 2 and level 3 BLAS are available. But the linear algebra chapter contains some routines for solving sparse linear systems by direct or iterative methods.

Fujitsu/Siemens. SSL II and supplements Fujitsu's SSL II library is supplemented by Siemens with a complete set of all three levels of BLAS, a set of FFT routines and random number generators. Together with these supplements SSL II seems to be the most comprehensive library from a supercomputer manufacturer. The linear algebra chapter contains several routines for solving linear systems and eigenvalue problems. There are routines for nonlinear equations, minimization, interpolation and approximation, Fourier and Laplace transforms, differentiation and quadrature, ordinary differential equations, special function approximation and random number generation.

2.2 Standard Numerical Libraries

While the manufacturer supplied subprogram libraries are available only on specific computer systems, standard mathematical subroutine libraries like IMSL/MATH library from IMSL Inc. and NAG Fortran library from the Numerical Algorithms Group Ltd. are available on a wide range of computer systems ranging from PC's to supercomputers. So program development and small production runs can be done on a workstation while large production runs are executed on a supercomputer. The identical program can be executed in different supercomputer environments.

Versions of these libraries are available for the most important supercomputers like CRAY Y/M, CRAY 2, IBM 3090 VF, NEC SX-series or Fujitsu/Siemens VP-series.

These libraries have not been designed for use on supercomputers from their starting point. But during the last years IMSL as well as NAG have spent much work in adapting the libraries to achieve efficient implementations on supercomputers. This must be done while maintaining the portability of the library, i.e.

- The user interface of library routines must not be changed. But the user interface of new routines which are introduced into the library may be chosen to allow efficient vectorization.
- The source code of library routines should be identical on all computer systems as far as possible. This is necessary...
in order to guarantee the quality of the software and to maintain the software over a long life cycle.

Both libraries, IMSL as well as NAG, have a much broader content than the libraries from supercomputer manufacturers. They comprise programs for solving systems of linear or nonlinear equations, eigenvalues system analysis, interpolation and approximation, integration and differentiation, differential and integral equations, Fourier and Laplace transforms, linear and nonlinear optimization, special function approximation and a lot of utility functions. Besides this, many statistical capabilities for data analysis are included. The routines from linear algebra chapters in most cases handle dense matrices, and a few routines in NAG library are available for sparse matrices, and IMSL library contains two routines for iterative solution of linear systems which operate on sparse matrices.

3 Adapting Libraries for Supercomputers

Work on adapting standard numerical libraries to the architecture of supercomputers must concentrate on those areas where most computing time is spent. These are linear algebra, FFT and solution of differential equations. An overview of vectorizing NAG and IMSL library is given in [3] resp. [7].

Besides restructuring of existing library routines new programs have been introduced into the libraries, especially into the NAG library, which are well suited for vectorization. So new routines for numerical quadrature and solution of differential equations have been incorporated which evaluate function values at many grid points in one subroutine call. Calling an external subprogram to evaluate only one function value would disturb vectorization and would increase the CPU time significantly. Also new routines for FFT, doing several transforms in parallel have been included. This gives in general greater vector lengths and higher speedups on vector computers.

In order to improve the efficiency of library routines on supercomputers a hierarchical programming concept must be applied. It is necessary to identify the most time consuming parts of programs which are general enough to be put into a library subset that can be modified and tuned for a specific supercomputer.

3.1 Linear Algebra

Basic Linear Algebra Subprograms (BLAS) have been defined in [6, 1, 2] and are widely recognized as a standard. Level 2 BLAS contain routines for matrix-vector-multiplication of rectangular, triangular, symmetric or hermitian matrices, rank-one- and rank-two-updates of rectangular, hermitian or symmetric matrices and solution of linear systems with triangular coefficient matrices. Matrices may be stored in different storage modes (general, banded or packed). Level 3 BLAS contain similar routines for matrix-matrix-operations: matrix-matrix-multiplication of rectangular, symmetric, hermitian or triangular matrices, rank-k- and rank-

2k-updates of hermitian or symmetric matrices and solution of linear systems with triangular coefficient matrices and multiple right hand sides. Carefully optimized versions of these routines are being supplied by most supercomputer manufacturers. Usage of these optimized BLAS will result in portable and very efficient software.

In the current version of IMSL- and NAG-library linear algebra routines are based on level 2 BLAS and a few routines already use level 3 BLAS. In the next releases level 3 BLAS will be used more intensively.

Table 1 shows the performance of some library routines for solving a system of linear equations based on LU factorization. L2TRC/LFSRC from IMSL, F04AAF from NAG, SGEFA/SCESL from SCILIB, DLAX from SSL II and DGEF/DSEL from ESSL. In all cases the array containing the matrix has been defined with an odd leading dimension in order to minimize memory bank conflicts.

### Table 1

<table>
<thead>
<tr>
<th>Computer</th>
<th>Library</th>
<th>matrix size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>100</td>
</tr>
<tr>
<td>Siemens</td>
<td>SSL II</td>
<td>155</td>
</tr>
<tr>
<td>S 400/10</td>
<td>NAG</td>
<td>147</td>
</tr>
<tr>
<td></td>
<td>IMSL</td>
<td>120</td>
</tr>
<tr>
<td>CRAY Y-MP</td>
<td>SCILIB</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>NAG</td>
<td>104</td>
</tr>
<tr>
<td></td>
<td>IMSL</td>
<td>114</td>
</tr>
<tr>
<td>IBM 3090 S</td>
<td>ESSL</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>VF</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>IMSL</td>
<td>34</td>
</tr>
<tr>
<td>CRAY 2</td>
<td>SCILIB</td>
<td>134</td>
</tr>
<tr>
<td></td>
<td>NAG</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>IMSL</td>
<td>24</td>
</tr>
</tbody>
</table>

Table 1: Performance of LU-factorization in MFLOP/s

Table 2 shows similar results for Cholesky-factorization, using LDFLD/LFSDS from IMSL, F03AEF/F04AGF from NAG, SPOFA/SPOSOL from SCILIB, DVSIX from SSL II and DPPE/DPPS from ESSL. Again the leading dimension of the array is odd except DVSIX and DPPE/DPPS which accept the matrix in packed storage mode in a one-dimensional array.

### Table 2

<table>
<thead>
<tr>
<th>Computer</th>
<th>Library</th>
<th>matrix size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>100</td>
</tr>
<tr>
<td>Siemens</td>
<td>SSL II</td>
<td>144</td>
</tr>
<tr>
<td>S 400/10</td>
<td>NAG</td>
<td>96</td>
</tr>
<tr>
<td></td>
<td>IMSL</td>
<td>99</td>
</tr>
<tr>
<td>CRAY Y-MP</td>
<td>SCILIB</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>NAG</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td>IMSL</td>
<td>119</td>
</tr>
<tr>
<td>IBM 3090 S</td>
<td>ESSL</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>VF</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>IMSL</td>
<td>26</td>
</tr>
<tr>
<td>CRAY 2</td>
<td>SCILIB</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>NAG</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>IMSL</td>
<td>40</td>
</tr>
</tbody>
</table>

Table 2: Performance of Cholesky-factorization in MFLOP/s

In order to study the behaviour of these routines in case of badly dimensioned arrays, the same programs have been executed on Siemens S 400/10 and CRAY Y MP for matrices of size 256 and 512 with different leading dimensions. The results are given in table 3 and indicate that the right selection of leading dimensions in the calling program can increase the performance significantly. Similar results as those reported in tables 1, 2 and 3 can be obtained for other linear algebra routines showing that the performance of routines from standard numerical libraries like IMSL and NAG
C. Ven to linear algebra and FFT. Only a small range of applications, most attention has been given to linear algebra and FFT. This demonstrates that well defined and optimized low level routines, in this case the BLAS, allow the development of very efficient and portable software. There is a need for comparable basic routines in other areas. In [4] a first idea for a set of Basic Operations for Fourier Transforms (BOFT) is given which hopefully will play a similar role for applications of FFT. Also discussions on sparse level 2 BLAS defining matrix-vector-operations for sparse matrices are going on. As with the BLAS these low level routines should become part of the software environment of a supercomputer. This is the basis for development of numerical libraries and application programs.

Today there are some open problems concerning numerical libraries on supercomputers:

- The application programmer must very carefully select the appropriate routine. Sometimes there are several library routines doing similar operations but using different data structures which may influence the performance significantly.

- When passing multidimensional arrays the leading dimension should always be an odd number. But some library routines do not include this as a separate item in the parameter list.

- Within one library the performance of different programs may vary significantly.

- Some important application areas (e.g. solution of sparse linear systems or sparse eigenvalue problems) are not covered by library routines.

### References


---

**Table 3: Effect of leading dimension**

<table>
<thead>
<tr>
<th>N</th>
<th>FFTVPLIB</th>
<th>NAG</th>
<th>IMSL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DFTCBl</td>
<td>C00FRF</td>
<td>DF2TCF</td>
</tr>
<tr>
<td>64</td>
<td>0.029</td>
<td>0.018</td>
<td>0.176</td>
</tr>
<tr>
<td>128</td>
<td>0.039</td>
<td>0.056</td>
<td>0.209</td>
</tr>
<tr>
<td>256</td>
<td>0.041</td>
<td>0.063</td>
<td>0.225</td>
</tr>
<tr>
<td>512</td>
<td>0.054</td>
<td>0.103</td>
<td>0.336</td>
</tr>
<tr>
<td>1024</td>
<td>0.092</td>
<td>0.171</td>
<td>0.531</td>
</tr>
<tr>
<td>2048</td>
<td>0.154</td>
<td>0.382</td>
<td>0.922</td>
</tr>
<tr>
<td>4096</td>
<td>0.322</td>
<td>0.740</td>
<td>1.698</td>
</tr>
<tr>
<td>8192</td>
<td>0.656</td>
<td>1.495</td>
<td>3.382</td>
</tr>
<tr>
<td>16384</td>
<td>1.468</td>
<td>1.682</td>
<td>6.652</td>
</tr>
<tr>
<td>32768</td>
<td>2.148</td>
<td>2.121</td>
<td>9.966</td>
</tr>
<tr>
<td>65536</td>
<td>2.865</td>
<td>12.216</td>
<td>12.960</td>
</tr>
<tr>
<td>131072</td>
<td>5.156</td>
<td>24.585</td>
<td>25.874</td>
</tr>
<tr>
<td>262144</td>
<td>10.636</td>
<td>50.105</td>
<td>53.865</td>
</tr>
</tbody>
</table>

Table 4: Timing of different FFT-routines on Siemens S 400/10

---

4 Conclusion

Subroutine libraries from supercomputer manufacturers cover only a small range of applications, most attention has been given to linear algebra and FFT. But in the field of linear algebra standard numerical libraries give equivalent or even better performance. Today there are some open problems concerning numerical libraries on supercomputers:

- The application programmer must very carefully select the appropriate routine. Sometimes there are several library routines doing similar operations but using different data structures which may influence the performance significantly.

- When passing multidimensional arrays the leading dimension should always be an odd number. But some library routines do not include this as a separate item in the parameter list.

- Within one library the performance of different programs may vary significantly.

- Some important application areas (e.g. solution of sparse linear systems or sparse eigenvalue problems) are not covered by library routines.
Supercomputers and Distributed Applications: Status, Trends, Needs
Klaus F. Hanauer and Andreas Knocke
Rechenzentrum der Universität Karlsruhe
Postfach 6980, D-7500 Karlsruhe 1, Germany

Abstract: The supercomputer in combination with the graphics super-workstation opens a new simulation technique. The scientific visualization enables computational scientist and engineers to undertake "human-in-the-loop" problems - a class requiring visualization techniques as different from photorealistic computer graphics as interactive computing is from batch processing. The developer of the necessary software systems that provide true "distributed applications" has to divide the application program to the existing resources - supercomputer, network and graphics super-workstation.

1. Introduction

The new series of powerful supercomputers, a new class of powerful workstations and the high speed LAN's in combination lead to a new class of applications called "distributed applications". The "graphics superworkstations" are expected to play an increasingly important role in providing an enhanced environment for supercomputer users.

Their potential uses include:

1. Off-loading the supercomputer
   - service station for the supercomputer as front-end system (input queue - output queue - print queue management)
   - pre- and postprocessing of the input and output of supercomputer applications
   - distributed or shared processing.

2. Scientific visualization
   - understanding of results
   - communication of results.

3. Real-time interaction with the supercomputer
   - controlling of iterative computations
   - kill, suspend and restart of supercomputer jobs
   - exploration and development of new algorithms.

2. Status

The term "graphics super-workstation" is defined here to refer to a category of workstations introduced in 1988 which combine high quality graphics with very powerful computational capability. Typically, such workstations provide from 1/10 to 1/100 the floating point speed of the most powerful current supercomputers, they have large main memories (16 - 256 MBytes) and are capable of generating and manipulating realistic, three-dimensional graphic displays. In combination with high speed LAN's it is possible to exchange data very fast between the different computers in the LAN and the workstation memory. The workstation provides a very rapid movement of data between memory, disk storage, computational units and graphics hardware. The extensive system and application software provides users with a powerful and convenient working environment. In the following table typical graphics and display manipulation characteristics will be specified. In the second column of table 1 the actual capabilities of the workstation installed in the computing center of the University of Karlsruhe will be presented.

<table>
<thead>
<tr>
<th>Typical Graphics</th>
<th>Workstation of the University of Karlsruhe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Screen display</td>
<td>19&quot; color monitor resolution 1280-1024</td>
</tr>
<tr>
<td>Image bit-planes</td>
<td>16-32 bit Z-buffer 24 bit color planes</td>
</tr>
<tr>
<td>Display speed</td>
<td>500,000 3D rendering 150,000 point shading</td>
</tr>
<tr>
<td>Surface geometry</td>
<td>polygons, triangular strip meshes, NURBS</td>
</tr>
<tr>
<td>Shading, lighting, rendering</td>
<td>Flat+, Phong+, Gouraudshading, textures, transparency, specular highlighting, ray tracing</td>
</tr>
<tr>
<td></td>
<td>Flat+, Phong+, Gouraudshading, radiosity, advanced hardware, lighting, transparency, ray tracing (optional)</td>
</tr>
</tbody>
</table>

Table 1: Typical graphics capabilities and values of the university workstations

Table 2 shows the typical values of a graphics super-workstation hardware and in the second column the corresponding values of the university machines.

The supercomputer performance, mainly the very fast floating point pipelines with a peak performance of 5 GFlops and the large main memory of 2 GByte is the reason that we need powerful peripheral workstations with super graphic capabilities to prepare the supercomputer results or to steer the simulation. Supercomputing is currently in the gigaworld era. Unfortunately, we may also be confronted with very large amounts of input data, but also as output data from solutions to mathematical equations representing physical, chemical or technical processes.

At the third IFIP International Conference on Data Communication Systems and their Performance an empirical ratio of approximately 100 Bytes of output per MFlops of calculation was found. Hence, as we approach processing power of 5 GFlops of our Siemens S600/20 vectorcomputer, and a simulation time of 2000 seconds, this empirical ratio would predict the following output:

\[
100 \text{ Bytes/MFlops} \times 5 \times 10^6 \text{ MFlops/sec} \times 2000 \text{ sec} = 1 \text{ GByte}
\]
Structural analysis or computational fluid dynamics are typical engineering problems requiring a vector computer for the solution of the discretized system using the finite element method employed in many commercially available codes as ADINA, FIDAP or LS DyNA 3D. An example of graphical output is shown in figure 1. The 3D turbulent flow around a car has been computed on the Siemens/Fujitsu S600/20 supercomputer using FIDAP and the results are visualized on the workstation.


<table>
<thead>
<tr>
<th>Typical graphics workstation</th>
<th>Workstation of the University of Karlsruhe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of processing units, floating point units</td>
<td>1-4 vector floating point pipes</td>
</tr>
<tr>
<td>Vector or floating point performance</td>
<td>5-30 MFlops</td>
</tr>
<tr>
<td>Integer processing</td>
<td>10-80 Mips</td>
</tr>
<tr>
<td>Cache memory</td>
<td>1 MByte (1 GByte/sec)</td>
</tr>
<tr>
<td>Main memory</td>
<td>16-128 MByte (300 MByte/sec)</td>
</tr>
<tr>
<td>I/O channels</td>
<td>80-100 MBit/sec</td>
</tr>
<tr>
<td>Disc storage</td>
<td>300-2000 MByte</td>
</tr>
</tbody>
</table>

Table 2: Workstation Hardware

In the computing center we can have up to 10 jobs per day of this 1/2 hour type, so we produce data output in an order of 10 GByte/day. These types of jobs are large scale calculations in the fields of finite element structural analysis, fluid dynamics involving repeated iterations over a spatial grid, and ab initio computational chemistry involving determination of eigenvalues of very large sparse matrices and multi-dimensional integrations.

The driving sources for increased processing speed, main memory and disc storage are the physical realism, the increased dimensionality and the data volume. Realistic representation of physical or technical systems may increase geometric complexity or eliminate simplifying approximations. On the other side, the dimensionality of a problem is not limited to the physical dimensions and the time, but more generally represents the number of degrees of freedom that must be considered as in the number of grid points in a computational fluid dynamics problem or the number of finite elements in a structural analysis problem.

Scientific problems of the departments of the University of Karlsruhe that are driving forces for the increase in computational power are:

- Computational Fluid Dynamics (including turbulence)
- Structural Analysis (finite elements, nonlinear analysis, different materials, eigenvalues, modification of the geometry)
- Physics/Chemistry (molecular dynamics, ab initio quantum chemistry, surface chemistry, statistical mechanics, astrophysics)
- Material Science (superconductivity, sinter materials, materials by design)
- Seismology
- Climate- and local weather simulation (environmental influences)

Generating the spatial discretization into different elements is done as preprocessing using a workstation freeing the supercomputer for the numerical intensive part of the analysis.

3. Graphics Super-Workstation in a Supercomputing Environment

The supercomputer of the University of Karlsruhe is needed to solve important scientific problems. The supercomputers of the future are needed to solve problems that presently cannot be done at all and should be designed and used for this purpose. The new generation of powerful workstations of the university provide a logical, cost-effective and user-time-effective alternative to shared supercomputers and indeed this is one of their appropriate and important roles in a supercomputing environment shown in figure 2. The increase in computational capability should not be the result of the aggregate demands of many users each of whom may need only a small amount of supercomputer time.
Regarding the large amount of results and output data, the graphics super workstation can also play the role of a "filter". Only the necessary information to detect new phenomena arrives at the workstation screen. Along with processing speed and larger main memories, there is a need for increased storage and communication bandwidth. The present limitation of our LAN's of about 140 MBit/sec is restricted but adequate for now, however the existing WAN's (excluding the BELWUE link between the universities of Karlsruhe and Stuttgart) are totally inadequate.

The telecommunication bandwidth required by a supercomputer is mainly proportional to its CPU speed. Table 3 shows the required speeds for transfer of different items.

<table>
<thead>
<tr>
<th>application</th>
<th>required speed</th>
<th>present technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>text</td>
<td>9.6 kBit/sec</td>
<td>ISDN (64 kBit/sec)</td>
</tr>
<tr>
<td>X-Windows</td>
<td>20 kBit/sec</td>
<td></td>
</tr>
<tr>
<td>color graphics file transfer (FTP)</td>
<td>1-2 MBit/sec</td>
<td>Ethernet (10 MBit/sec)</td>
</tr>
<tr>
<td>NFS</td>
<td>&gt; 1 MBit/sec</td>
<td></td>
</tr>
<tr>
<td>simulation visualization (6 pictures/sec)</td>
<td>64 MBit/sec</td>
<td>FDDI (100 MBit/sec)</td>
</tr>
<tr>
<td>animation</td>
<td>1-10 GBit/sec</td>
<td>Ultrانet (800 MBit/sec), frame buffer</td>
</tr>
</tbody>
</table>

Table 3: Speeds for transfer of different items

The problem of mass storage is even more limiting (University supercomputer disc storage capacity ~ 40 GByte, compare with expected output for advanced simulations) and there are few promising technological developments on the horizon.

The only hope to overcome these problems is to find out a new way in which we make use of supercomputers so as to effect a drastic reduction in the amount of data that needs to be stored or transferred. This requires a fundamental new kind of scientific simulation technique. The emergence of graphics superworkstation offers an opportunity to enable that essential change in methodology. Today the typical solution sequence of a simulation includes the following steps:

- Physical System Description
- Mathematical Model
- Vector Algorithm
- Calculation
- Analyze Results
- Presentation of Results

The handling of these steps in a sequential top-down manner must be replaced by a new simulation technique using the capabilities of the graphic superworkstations. Graphics superworkstations can be utilized further "upstream" in the process as part of the calculation itself. Their specialized hardware and architectural properties can be used effectively in conjunction with the supercomputer in a distributed processing system.

However, the availability of X-windows on workstations and supercomputers not only enables a user to watch the progress of a computation - and if necessary abort a bad run - but permits the user to interact with the computation in process. The scientist can modify parameters, such as step size, grid spacing, damping terms, etc and also change the solution algorithm. Human interaction with the supercomputer by means of powerful graphics superworkstation will also enable or facilitate the solution of computationally difficult problems where the intervention of a human is a key or possibly essential part of the simulation. Examples include algorithms with different data storage techniques, systems containing multiple extrema where global extremum is desired and iterative processes that converge very slowly.

In these examples, the scientist becomes an essential part of the simulation. The user acts at a very high level by setting as part of the calculation nonlinear feedback loop, by using the knowledge of physical behavior not included in the program code or by detecting trends.

4. Technical aspects of "Distributed Applications"

The technical realization of distributed applications must be based on the existing hardware systems, the not yet fully standardized networks, the different operating systems and the different graphical standards. Standardization is a difficult task involving quite a number of organizations (ISO, IEEE, CCITT, ANSI, DIN), competing manufacturers and specialists and will take time and an still evolving market. The requirements of the parts of the distributed system are shown in table 4.

<table>
<thead>
<tr>
<th>supercomputer</th>
<th>communication services</th>
<th>workstation</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIX System V</td>
<td>remote login</td>
<td>operating system: UNIX System V, Berkeley Extens., OSF/1</td>
</tr>
<tr>
<td>Berkeley Extens.</td>
<td>remote job entry</td>
<td>networks: Ultrанet FDDI Ethernet</td>
</tr>
<tr>
<td>High speed I/O</td>
<td>remote monitoring</td>
<td>languages: FORTRAN, C, C++ (object oriented)</td>
</tr>
<tr>
<td>HIPPI</td>
<td>remote printing</td>
<td>graphiсal standards: X11, Motif, GIF, PCX, CGM</td>
</tr>
<tr>
<td>FDDI</td>
<td>file transfer (ftp)</td>
<td>different picture interchange formats: IGES, GIF, PCX, CGM</td>
</tr>
<tr>
<td>Ethernet</td>
<td>connecting</td>
<td></td>
</tr>
<tr>
<td>Channels (Hyper, IBM)</td>
<td>distributed processing</td>
<td></td>
</tr>
<tr>
<td>(TCP, UDP/IP)</td>
<td>remote procedure calls (RPC)</td>
<td></td>
</tr>
<tr>
<td>auxiliary services</td>
<td>remote windowing</td>
<td></td>
</tr>
<tr>
<td>network computing (NCS)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Technical requirements

5. Conclusion

Today's supercomputers produce torrents of data, but the human brain still interprets numerical data as poorly as it always has. Scientists and engineers need an alternative to numbers and that alternative is images.

The overview of the characteristics and capabilities of currently available graphics superworkstations showed that one of the most important activities in a supercomputing environment was the analysis and interpretation of large masses of complex information.

The role of the graphics superworkstation in a supercomputing environment is absolutely essential to ensure the integrity of analysis, to provoke insights and to empower the human as an essential component of the simulation.
Abstract. We address the problem of benchmarking supercomputers with the aid of synthetic programs. While not modelling an actual workload this approach has the advantage of providing more general information about the capabilities of the systems under consideration. We will review some benchmarking models and present the EuroBen benchmark as an example of a benchmark set that yields a performance profile which enables the identification of the strong and weak points of a machine in terms of a range of applications and their constituent algorithms.

I. INTRODUCTION

In this contribution we focus on the performance of supercomputers. A problem encountered with these systems is the large variety in their architecture. These can range from large vector computers with a limited amount of processors that share a common memory to machines with thousands of very simple processors and distributed memories. This can lead to an enormous performance range, sometimes potentially a factor of thousand or more, depending on the suitability of a certain piece of code for the underlying architecture.

There are several ways to deal with this problem, leading to different ways of benchmarking. Three main ways can be identified: Theoretical benchmarking, Synthetic benchmarking, and Practical benchmarking [12]. We will briefly discuss the various approaches and their particular advantages and drawbacks in the next three sections. In addition, in section V we will discuss one synthetic benchmark, the EuroBen benchmark, in more detail.

II. THEORETICAL BENCHMARKING

Theoretical benchmarking is aimed at the modelling of the performance behaviour of machines with regard to their constituent components. These machines may or may not exist. In fact, this kind of modelling is often used to estimate trade-offs between performance and the application of more or different hardware components, more or less expensive technology, etc. Mostly, probabilistic models are used to estimate the influence of the various components. As supercomputers tend to be more complex than their more conventional counterparts the composition of theoretical benchmarks also is much more difficult and one often has to rely on simulators to acquire the desired information. This branch of benchmarking is more or less becoming a discipline in itself with journals like Performance Evaluation as an example of the interest in this field and a growing number of publications in other journals (see for instance [1], [5], [8], [11]).

An obvious advantage of the theoretical approach is that it is often able to simulate machine- or component behaviour at a fraction of the time and price of actually building such a system. For existing machines a correct theoretical benchmark may adequately predict upperbounds and even performance profiles for various patterns of utilisation of the components.

A drawback is the difficulty of designing good theoretical benchmarks, especially for very complex behaviour large multi-user supercomputers. In most cases one has to be satisfied with the limited understanding obtained for isolated subsystems without being able to relate this to the total performance of the system. In addition, relatively small changes in the architecture of a machine may be difficult to incorporate in the machine model to be evaluated. In all, theoretical benchmarking is a very difficult (but sometimes rewarding) approach to performance evaluation.

III. SYNTHETIC BENCHMARKING

Synthetic benchmarking consists of the running of one or more program kernels from which the performance of a system should be derived. The practice of running such synthetic programs is very easy in comparison to theoretical benchmarking or the practical benchmarking to be discussed later. This, and the fact that many "ready-made" benchmarks exist ([3], [10]), has made it very popular. Yet, although easy to run, the interpretation and the significance of the results for a particular test site are often far from straightforward [4, 6]. Especially with supercomputers one has to be very careful in addressing all aspects of the machine(s) at hand and one should refrain from judging such systems from just one parameter which supposedly would represent the performance of the system. Regrettably, this is a common practice with many vendors which in this way are able to "prove" the relative superiority of their systems. We will discuss better synthetic procedures in section V.

The advantages of synthetic benchmarking are already made clear. The testing procedure is relatively easy (although the design of a good synthetic benchmark is not). Synthetic benchmarking also is flexible. When new architectures emerge one can modify or extend an existing benchmark set easily to address new architectural features. In addition, when performed with care, the benchmark may yield more general information than just performance figures for particular programs.

Apart from the danger of misinterpretation that al-
ready has been mentioned, synthetic benchmarks cannot claim to give definite answers in selection procedures for a particular site. In this case one has to do additional benchmarking that more precisely reflects the workload of that site and possibly the variations in and evolution of this workload. So, although synthetic benchmarking may help in the preselection stage and in acquiring a general understanding, one has to complement it with site-specific testing procedures.

IV. PRACTICAL BENCHMARKING

Practical benchmarking is employed to obtain the specific answers concerning a systems appropriateness for a particular site. In this case a typical workload for that site can be run on such a system and usually one already has a detailed knowledge about configuration requirements and the total software environment that has to present. Because of the very specific nature of these benchmarks it is very hard to make general statements about the methodology beyond the most trivial ones: one should work with a representative workload, memory- and I/O-requirements should be properly heeded, etc. Consequently, there is almost no literature of general interest on practical benchmarking (a rare example is [9]). The large architectural differences between supercomputers will often frustrate practical benchmarking because it is impossible to port a complete workload to these systems in a way that allows simple comparison. This leads to the paradoxical situation that where a clear comparison is possible to port a complete workload to these systems in a way that allows simple comparison. This leads to the paradoxical situation that where a clear comparison is the correct way to summarize benchmark results, Comm. of the ACM, Vol. 39, No. 6, 1986, 218-221.


REFERENCES

SUPERCOMPUTERS: THE SOFTWARE, VECTORIZING AND PARALLELIZING COMPILERS

Karin Roger Kast
Rechenzentrum der RWTH Aachen
Seffenter Weg 23
D-5100 Aachen
Germany

Abstract - Recently, the biggest increase in performance - measured in "naked" IPS or FLOPS - had been achieved by hardware developments and new machine architectures. That is why today the main attraction often is the architecture of new systems. In case of multiprocessor systems new results will be expected soon. Another aspect which becomes more and more important is the quality of software. For economical reasons standard software, e.g. the FORTRAN programming language, is preferred rather than special architecture dependent or assembler languages. The benefits are easier implementation, better portability and the possibility to use universal programs and libraries on different machines. Portability for example is very important since the life cycle of application software lasts longer than the innovation time of hardware. Consequently the FORTRAN programming environments for some supercomputers from CRAY, IBM and Fujitsu are presented. The optimizing strategies of the compilers for vectorizing and parallelizing will be discussed but no valuation of the efficiency of the generated code will be made because the strongly depends on the underlying hardware. Besides, some language extensions, compiler-directives and analysing tools are presented. Operating systems will be focussed briefly since they build the missing link between runnable programs and hardware.

1. INTRODUCTION

After a general survey about the operating systems for supercomputers from CRAY, Fujitsu and IBM the compiler systems CRAY CFT 77, Fujitsu FORTRAN/VP and IBM FORTRAN/VS are presented and compared in detail. The machines from CRAY and the IBM 3081 series are multiprocessor computers tightly coupled via shared memory. The older Fujitsu VP machines are monoprocessor machines but the new VP machines have a multiprocessor option and enhanced vector units. All machines are vector computers with some possibilities for parallelism provided by the shared memory. Although CFT 77 and IBM VS have some multtasking capabilities, their main goal is to vectorize rather than parallelize programs. The Fujitsu VP compiler has no multitasking features but a new one is announced which will support the new multiprocessor architecture.

In addition to these common FORTRAN systems that are extensively used in production environments two recent developments with better support for parallelism will be presented. CRAY's autonomous system CFT 77 and IBM's PARALLEL FORTRAN. Both systems have capabilities to parallelize work automatically and powerful extensions for explicit parallelism.

Vectorizing and parallelizing may be regarded as program transformations which change the order of statements but don't change the program's semantics. If two statements share some input or output variables - e.g. a data dependency - they must not be interchanged. Therefore the autovectorizing or autoparallelizing compilers analyse these dependencies before they change the order of statements. There are similar analysing and transformation methods for parallelization and vectorization.

2. OPERATING SYSTEMS

On IBM 3090 mainframes there are two major multiuser operating systems: MVS and VM. MVS is a multi-purpose operating system which has a wide functionality and is used in scientific and commercial environments as well, whereas VM is commonly used in scientific environments.

MVS (Multiple Virtual Storage) is designed to let each user have his own addressing space which is extremely protected against disturbance from outside of the user's world. Each user may start several tasks that are executed concurrently or in parallel. System resources are managed dynamically by MVS. Thus, MVS supports parallelism in an efficient way.

VM virtualizes system resources. Each user has his own virtual machine (VM) which is separated from other ones. But, anyhow, the user may initiate communication with other VMs. These VMs send their requests to a control program (CP) which in turn carries out the requested operations. Some VM's normally are dedicated as batch worker machines to realize batch processing. Parallelism is supported by assigning several VMs to a user application that are working independently.

VSP/I (Vector System Programming Interactive) for Fujitsu's vector machines is largely compatible with MVS/XA (user and operator interface, job control but not binary code). VSP/I has some extensions concerning vector processing on these special machines. For instance, programs under MVS usually operate in a virtual addressing mode but in VSP/I all vector programs run in a real mode. The whole program code is resident in main storage. Hence the vector processor never waits on program parts to be loaded from secondary storage.

The newest operating system for CRAY supercomputers is UNICOS which is based on AT&T UNIX SYSTEM V. UNICOS has some enhancements to UNIX for mainframe supercomputers, for example batch services, accounting tools and performance analysis tools. Above all UNICOS supports the multiprocessor architecture of CRAY machines.

3. STANDARD FORTRAN PROGRAMMING ENVIRONMENTS

3.1 THE COMPILERS CRAY CFT 77, FUJITSU VP, AND IBM VS

Now we take a closer look at the compilers CRAY CFT 77, Fujitsu VP, and IBM VS. All these compilers are autovectorizing compilers based on FORTRAN 77 standard. There are language extensions with CFT 77 modelled in FORTRAN 8X standard. Information that cannot be extracted by syntactical analysis, for example vector lengths and true ratios of logical IF statements may be inserted as compiler directives into the source text. These directives will be treated like comments by other compilers. The compiler listing can be helpful when optimizing a program. All compilers show informations about the optimizations of loops, furthermore the IBM VS compiler presents all loops in a source to source modification. Hence this listing could be a vectorizing tool for other systems. Tools for run time analyses are available with all systems except IBM VS under VM/VSX. The most comfortable is VECTUN from Fujitsu which allows interactive analysing on loop and statement level. Optionally the results of this analysis may be taken to generate compiler directives automatically.

<table>
<thead>
<tr>
<th></th>
<th>CFT 77</th>
<th>Fuj. VP</th>
<th>IBM VS</th>
</tr>
</thead>
<tbody>
<tr>
<td>auto-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vectorizing</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>auto-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>parallelizing</td>
<td>cf 77</td>
<td>planned</td>
<td>-</td>
</tr>
<tr>
<td>programmed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>multitasking</td>
<td>X</td>
<td>planned</td>
<td>X</td>
</tr>
<tr>
<td>vector</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>language</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>extensions</td>
<td>8X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>directives</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>analysis</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tools</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>X</td>
<td>X</td>
<td>only MVS</td>
</tr>
</tbody>
</table>

8X = FORTRAN 8X standard
CF77 = possible if CFT 77 is embedded in cf77
Table 3.1: Compiler equipment
3.2: OPTIMIZING METHODS OF AUTOVECTORIZING COMPILERS

The quality of vectorizing and parallelizing compilers depends decisively on the ability of analyzing and modifying loops. Nevertheless, loops with certain data dependencies after some optimization can be vectorized. For instance, the loops with the data dependencies in Table 3.2.1. can be vectorized if the statements are exchanged.

```
DO 10 I=1,N
    A(I) = ...
A(I+1) = ...
10 CONTINUE
```

Table 3.2.1.: data dependence, anti-dependence

For IF statements with an anti-dependence that cannot be reordered another vectorizing technique may be chosen. A temporary scalar is introduced which is expanded to a vector by the compiler (see Table 3.2.2)

```
DO 10 I=1,N
    S = A(I+1)
    A(I) = ...
    S = ...
10 CONTINUE
```

Table 3.2.2.: anti-dependence after insertion of a temporary scalar

If a loop contains vectorizable and non-vectorizable statements vectorizable parts have to be separated from non-vectorizable to vectorize a portion of the loop. This is called partial vectorizing. Sometimes it depends on the value of a variable whether a loop is vectorizable or not. With the conditional vector method it is possible to generate sequential and vector code and to decide at run-time which code is executed. In nested loops statements of the outer loop may be independent from those of inner loops. Then outer and inner loops may be split into two loops which are both candidates for vectorization (loop distribution). The vectorization of an outer loop is attractive if the inner loop is not vectorizable or the vectorization of the outer loop is simply more efficient, for example because of a greater vector length. This implies that there might be interchanged loops (loop innermost)

For a logical IF statement in a loop there are different methods for vectorization. If there are vector-bit-masks the conditions will be stored and only those right sides of assignments where the associated mask bit is true will be assigned to the left side. The disadvantage of this method is that all operands are sent through the pipeline, this would be wasteful if the IF condition seldom is true. In this situation the gather/scatter technique works most efficiently. For all true conditions an index vector will be built and a special gather/scatter hardware accesses the operands indirectly with this vector and evaluates the right sides of equations. With the comp.ass. method a vector-bit-mask is built as well but all operands are fetched from memory avoiding indirect addressing. Then the fetched vectors are compressed and processed by constant stride hardware. After processing the result vector will be expanded and stored. This method will work well if the ratio of load/store and arithmetical operations is small.

These described optimizing techniques for compilers are listed in Table 3.2.3.

<table>
<thead>
<tr>
<th></th>
<th>CFT 77</th>
<th>Fuj. VP</th>
<th>IBM VS</th>
</tr>
</thead>
<tbody>
<tr>
<td>statement reordering</td>
<td>cf 77</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>partial vectorizing</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>temporary scalars</td>
<td>X</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>conditional vectors</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>vectorizing outer loops</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>loop distribution</td>
<td>-</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>logical IF</td>
<td>M</td>
<td>M,G,C</td>
<td>M</td>
</tr>
<tr>
<td>Block IF</td>
<td>G</td>
<td>M,G,C</td>
<td>M</td>
</tr>
<tr>
<td>ELSE IF</td>
<td>G</td>
<td>M,G,C</td>
<td>M</td>
</tr>
</tbody>
</table>

M = mask operation
G = gather/scatter operation
C = compress/expand operation
cf77 = possible if cf 77 is embedded in cf77

Table 3.2.3.: optimizing techniques

4. PARALLELIZING COMPILERS

4.1 GENERAL PARALLELISM CONCEPTS

Parallel execution of tasks for one application may improve its execution time but it increases the amount of the work to be done (in CPU cycles) because there is a remarkable overhead generating parallel tasks, their synchronization, and communication. Therefore, parallelism must be handled carefully. However, if an application covers most of the main storage it should be processed in parallel since it would be wasteful to execute it on only one CPU because of other CPU’s running idle. In general to maximize throughput the share of main storage and processing power should be equal. Another motivation for parallel processing may be that an application may be done only in realistic time by parallel speed up or the application has a certain deadline, e.g., meteorological calculations.

There are two classes of multithreading concepts:

- Fine-grain parallelism means parallel execution of small fragments of code, for example different chunks of a DO loop
- Coarse-grain parallelism signifies parallel execution of program segments with larger granularity, for example subroutines

Up to now techniques for interprocedural data dependency analysis to allow automatic coarse grain parallelization are still in the experimental phase. The programmer has to introduce explicit parallelism by using a multithreading interface, for example library calls, or new language elements or by specifying compiler directives. All interprocedural data dependencies have to be synchronized explicitly by the programmer. Within one subroutine compilers may find small independent parts of code (fine-grain parallelism) Thus, no restructuring of the algorithms will be needed. In coarse-grain parallelism it is a big problem to partition work into portions of approximately equal size so that the work load is balanced among the processors. Fine-grain parallelism helps load balancing by filling the gaps of processor idle time owing to the existence of many small computational units. If the sum of idle times of processors is greater than that for
Autotasking is unpredictable order. It must therefore remain independent. Data dependencies may be
processes within one MCS are scheduled dynamically in unpredictable microtasking is very efficient. Within one MCS several parallel processes are rescheduled and are immediately ready to work again. Therefore, busy loop concept into account. Logical CPU's which terminate working are able to execute

4.2. CRAY AUTOTASKING SYSTEM cf77

CRAY provides three concepts for multitasking: macrotasking, microtasking and autotasking. As macrotasking is the coarse-grain mechanism, microtasking and autotasking are fine-grain. With autotasking automatic fine-grain parallelization in subroutines is introduced. Autotasking, microtasking and macrotasking can coexist in different subroutines but not in a single one.

MACROTASKING

Macrotasking is available by means of FORTRAN subprogram calls from a special multitasking library and offers high level synchronization and communication primitives. These subroutines are the interface to the library scheduler which manages the single tasks, the synchronization between them and performs requests to the operating system. The library scheduler does not use hardware for most synchronization primitives. Thus for small granularity the activities of the library scheduler can lead to an extensive overhead. Macrotasking should be used with large granularity.

THE MACROTASKING LIBRARY consists of:

- routines to manipulate tasks
- routines to control events for synchronization
- routines to control critical regions which can be executed by one processor exclusively at one time

MICROTASKING

Microtasking offers much faster basic functions for synchronization and communication by accessing special hardware and is realized by compiler directives (CMICS). The CMICS-directives are translated into multitasking library calls.

THE MICROTASKING DIRECTIVES consist of:

- directives requesting microtasking mode and logical CPU's
- directives to define microtasking control structures (MCS)
- directives to mark critical regions

With the GETCPUS directive logical CPU's are created. Logical CPU's are in a busy loop waiting for work. If there is some work, to do all idle CPU's are able to execute parts of it. The operating-system scheduler takes this busy loop concept into account. Logical CPU's which terminate working are rescheduled and are immediately ready to work again. Therefore, microtasking is very efficient. Within one MCS several parallel processes may be defined each of them executed exclusively by one logical CPU. Processes within one MCS are scheduled dynamically in unpredictable order and therefore must be independent. Data dependencies may be considered by putting the corresponding processes into different MCS's, since a MCS must be terminated before the program continues. Code outside MCS's can be processed in parallel by all logical CPU's in unpredictable order.

AUTOTASKING

Autotasking is realized by using some microtasking and new autotasking.

CMICS-directives.

The CRAY autotasking system mostly consists of well-known components for instance the cf77 compiler. Other new parts are the preprocessor fpp and the multitasking translator fnp.

fpp analyses the data dependencies and transforms the source code to modified source code so it can be vectorized or parallelized or is simply more optimal in scalar. The modified source code contains directives for vectorizing and multitasking. Automatic parallelization may be switched off so fpp may create pure vector source code. Depending on the CRAY architecture, vectorization is preferred rather than parallelization. fpp only analyses DO loops, which are the most frequent type for parallel or vector constructs. All loops are vectorized if possible. In nested loops the outermost loop will be parallelized if possible and the innermost loop will be vectorized. By an option vectorizable loops with high iteration counts may be split into several parallel loops and each parallel loop will be vectorized. Even the outer loops of non-vectorizable inner loops may be parallelized. fpp provides inline-expanding of subroutines so a more global analysis of data dependencies can be made. But it should not be overdone because then the code grows immense. If no additional loops became vectorizable which should be the case in good vector programs the gain of run-time is little.

If no parallelism is requested, the modified source code program may directly be compiled with cf77. However, to exploit parallelism, the output of fpp must be processed by fnp before being translated with cf77. fnp changes autotasking directives to machine dependent library calls.

4.3 IBM PARALLEL FORTRAN

Parallel FORTRAN has three fundamental extensions for fine- and coarse-grain parallelism on IBM 3090 multiprocessors.

- extensions to the compiler for automatically generating parallel code
- language extensions for explicit parallelism
- extensions to the library for synchronizing parallel execution

Fine-grain parallelism is supported by language-extensions and automated detection of implicit parallelism, coarse-grain by language and library extensions.

The Parallel FORTRAN environment consists of the parallel application program, some FORTRAN processors and the parallel library. The FORTRAN library maps the parallel pieces of work onto virtual processors called FORTRAN processors. The operating system maps FORTRAN processors onto real machine processors. The implementation of FORTRAN processors depends on the operating system. FORTRAN processors under MVS/XA are tasks. Under VM/XA several virtual machines build a virtual multiprocessor machine and each FORTRAN processor is executed by one virtual CPU. The maximum degree of parallel execution can be varied at run-time by specifying an option for the number of virtual FORTRAN-processors. The user can define more parallel pieces of work than the number of FORTRAN processors allows. Additional pieces of work build a queue and finishing FORTRAN processors take their work from this queue. As long as the queues are filled the FORTRAN processors can continue working without the operating system. Only if a queue empties or refills there is overhead of the operating system by suspending or restarting a FORTRAN processor. There is no operating system overhead to schedule and synchronize parallel work.

AUTOMATIC PARALLELISM

The compiler determines whether it is cost effective to execute the loop in parallel, otherwise vector or scalar code is generated. A loop selected for parallel execution may contain inner loops that are parallelized, vectorized or serial. A loop selected for vector operation may contain only inner loops which are serial. There are for example directives to indicate a preference for parallel, vector or serial code and the number of iterations to be...
grouped together as a unit of work.

**LANGUAGE EXTENSIONS**

Parallel FORTRAN provides two types of language extensions which both can be nested and intermixed:

- **In-line extensions**: define parallelism within a routine
- **Out-line extensions**: define parallelism across routines

In general, in-line extensions support fine-grain parallelism as out-line extensions provide coarse-grain parallelism.

**IN-LINE EXTENSIONS**

There are two new in-line extension language elements: the PARALLEL LOOP and the PARALLEL CASES. Each iteration of a PARALLEL LOOP can be executed concurrently and the order of execution of the iterations is not guaranteed. Parallel executed blocks of statements are declared by PARALLEL CASES. Such blocks may contain straight code, parallel or vector loops. Although, PARALLEL CASES is an in-line construct this primitive should be better used for coarse-grain parallelism because in current implementations there is a large overhead involved with it. Thus, it is possible to number cases to define an execution order among blocks of statements. Every acyclic graph of data dependencies may be transformed into PARALLEL CASES.

**OUT-LINE EXTENSIONS**

Out-line extensions create disjoint, asynchronous execution environments called tasks. A task is a collection of subroutines that is independent of other tasks. This task concept is hierarchical, in the beginning there is one root task, which may create several child tasks with an ORIGINATE statement. A task is terminated with a TERMINATE primitive or automatically at its end. SCHEDULE or DISPATCH assign the execution of a subroutine to a task. As dispatched tasks run completely asynchronously scheduled tasks require a WAIT FOR instruction to terminate. Other primitives for subtask-synchronization are locks and events, which are recommended because explicit WAIT FOR is very expensive. Locks ensure that only one processor at a time gets access to a certain resource. Events may be used to make a task wait until another task has reached a certain point of execution. With locks and events self-balancing dynamic workloads can be created. PARALLEL LOOP and PARALLEL CASES constructs are automatically load balanced by the library. Only if the user explicitly shares data between tasks the execution of tasks can affect one another.

5. CONCLUSION

Today vectorizing is an established method. The three described compilers all are able to vectorize automatically but in spite of that they are very different. The CRAY CFT compiler is the successor of the older CFT 1.15 compiler. Since the aged CFT 1.15 was not based on theoretical program analysis, CFT has been designed completely new. The IBM FORTRAN/VS compiler is based on an older sequential version. The biggest handicap for VS is the underlying 3050 cache-hardware. If the accesses run out of cache-memory (for example row accesses in matrices) vectorizable loops are not vectorized by the compiler. The newest version of FORTRAN/VS Rel 2.5 is an autovectorizing and autoparallelizing compiler. Supplementary to enhanced vectorizing capabilities it includes features from PARALLEL FORTRAN. Fujitsu's VP compiler adapts the structure of vector registers to the program structure given. It has three vectorization methods for IF statements (mask, gather/scatter, compress/expand). The announced Fujitsu FORTRAN/VP-EX compiler will adapt programs to the new hardware and improve the performance of programs. For instance, the vector-pipeline is used inefficiently if the DO loop does not contain enough operations. Loop unrolling expands the executable statements in a DO loop and reduces the iteration count, thus, pipelines can work more efficiently. Program performance is increased further by integrating subroutines into calling routines and optimizing the modified program in a more global manner. To exploit the new multiprocessor architecture the compiler FORTRAN/PP will be developed with automatic intraprocedural parallelism and language extensions for interprocedural parallelism.

There is no ideal compiler, all are in continuous development and in addition to that often have to be adapted to new hardware.

With parallelization however new problems arise, for instance for coarse-grain parallelism synchronization has to be programmed explicitly. To justify the overhead derived from parallelization work has to be partitioned in large parts of code. This in fact conflicts with dynamic load-balancing. If the programmer is aware of his problem and of the underlying hardware too, many errors may be avoided. Hardware aspects are important since, for instance, the shared memory systems focussed here not only use memory to store data but also to communicate between tasks. Parallelizing compilers today are in the age of maturity. Possibly the next generation will be able to analyse data dependencies above the DO loop level. Analysing results will be displayed and some sort of interactive semi-automatic parallelizing will be realized.

6. ACKNOWLEDGEMENT

I would like to thank W. Jüng, who made valuable contributions to this paper.

7. BIBLIOGRAPHY


Dataflow Programming Languages

Rishiyur S. Nikhil
Laboratory for Computer Science
Massachusetts Institute of Technology
545 Technology Square, Cambridge, MA 02139, USA

Introduction

Dataflow languages are attractive for parallel computation because of their expressive power — their high-level, machine-independent, implicitly parallel notation — and because of their fine-grain parallelism, which seems essential for effective, scalable utilization of parallel machines.

Dataflow languages are no longer tightly coupled with dataflow architectures. Today, we understand the compiling issues for dataflow languages well enough that we can recommend these languages to programmers without having to rely on architectural support.

There are several dataflow languages described in the literature, including Id [7], Sisal [6], and Val [1]; due to space limitations, we will base our discussions here on Id.

Expressive Power of Dataflow Languages

"Expressive power" does not have a precise definition; one can make reasoned arguments about why one language is more expressive than another, but ultimately the judgment is subjective. Theoretically, FORTRAN and assembly-language have the same computational power, but almost everyone would agree that FORTRAN is more expressive, because it is architecture-independent and because it requires less detailed specification of problem solutions and is "closer" to the way human programmers think about problems.

Dataflow languages are based on functional programming languages. Unlike most programming languages that have evolved upwards from machine languages, functional languages are based on the λ-calculus, a mathematical theory of functions that predates digital computers [9].

Functional notation:
An aspect of the expressive power of functional languages is that one can manipulate complex objects (entire data structures, and even functions themselves) as ordinary values. Consider this function:

\[
\text{def map2 f A B} = \ldots \quad \text{details omitted} \ldots
\]

taking 3 arguments: a function \( f \) and two arrays \( A \) and \( B \), and returning as its result a new array (call it \( C \)) where

\[
\text{C}(i) = f \text{ A}(i) \text{ B}(i)
\]

i.e., it applies \( f \) to each pair of components of \( A \) and \( B \) and returns an array containing the results. (In Id and other functional languages, it is common to use a "curried" notation, omitting parentheses, commas, semicolons etc. around the arguments of a function.)

Now, we can specialize map2 to define a vector sum function:

\[
\text{def vsum A B} = \text{map2} (+) A B
\]

Here, the function \( + \) is supplied as the \( f \) parameter to map2, so that \( \text{vsum} \) adds the vectors \( A \) and \( B \) pointwise, and returns a vector containing the sums. Further, since \( \text{vsum} \) itself an addition function, we can perform vector addition on vectors whose components are themselves vectors:

\[
\text{def vsum A B} = \text{map2} \text{ vsum A B} ;
\]

Another useful higher order function is:

\[
\text{def foldl f z A} = \ldots \quad \text{details omitted} \ldots
\]

which computes the "\( f \)-reduction" of array \( A \), where \( z \) is the zero of the function \( f \), i.e., it computes:

\[
f \ldots (f (f (f (f (z) 1) 2) 3) 4) \ldots
\]

Using foldl, one can express the inner-product of two vectors:

\[
\text{def sp A B} = \text{foldl} (+) 0 (\text{map2} (+) A B) ;
\]

Here, \( \text{map2} \) multiplies \( A \) and \( B \) pointwise, and \( \text{foldl} \) computes the sum of the resulting vector of products.

The ability to manipulate entire complex objects and functions themselves as values relieves the programmer of the tedium of descending down to explicit loops and assignments at every stage — such details can be hidden in a few generally useful functions such as \( \text{map2} \) and \( \text{foldl} \) [2]. Such higher order functions are the "power tools" of functional programming, in the sense that they amplify the programmer's effort.

Implicit parallelism:

Another aspect of expressive power is implicit parallelism. In most parallel languages, the user must carefully orchestrate parallelism: explicitly partition the program into parallel processes, specify the placement of data, specify the placement and scheduling of processes etc. Often, these details are architecture-specific, resulting in fragile, non-portable code.

In functional languages (and in all our examples above) we do not have to specify what can be done in parallel. In principle, everything can be performed in parallel, limited only by data dependencies. For example, the implementation of \( \text{map2} \) may perform all the applications of \( f \) in parallel. Thus, functional languages do not increase the complexity of the programmer's task when moving from a sequential to a parallel environment.

M-structures: implicitly synchronized state:

In pure functional languages, there is no concept of "state," i.e., structures whose contents change during program execution. There are no assignment statements — programs are functions that compute new values from old. While many algorithms can be expressed clearly in this style, there are some problems.

One difficult area is algorithms making effective use of non-determinism. Functional languages have the Church-Rosser property [3] which guarantees that the answer produced by a functional program does not depend on the particular parallel execution structure chosen by an implementation. This is usually a major advantage; unlike other parallel languages, functional programs are not subject to race conditions leading to non-deterministic and irreproducible behavior (which complicates debugging).

However, consider the problem of traversing a directed graph to count the number of nodes reachable from a given root node. The following strategy seems simple: starting at the root node, fan out...
in-parallel on all outgoing edges from each visited node. To avoid repeated traversals of shared subgraphs (and cycles), the first traversal to reach a node marks it "visited," so that subsequent traversals reaching the node via other paths will observe the mark and retreat. Non-determinism is exploited because, of all the incoming traversals arriving at a shared node, we do not care which one arrives first to mark the node and traverse its successors. There is no way to express this idea in a functional language one must pick a deterministic order to traverse the nodes. This not only clutters up the program significantly, but also sequences it.

Introducing non-determinism into a functional language is not very different from introducing state—constructs for one can be used to simulate the other. Not surprisingly, functional languages also have approaches towards this computational notion. New compilers are unlikely, but also sequential. It initializes it. Some threads are waiting (for long-latency memory requests, for synchronization, for results of procedures, etc.).

Dataflow languages and their compilers offer a systematic, complete approach towards this computational model. New compilers are under construction at MIT and Berkeley that use dataflow principles to compile Id even for non-dataflow machines. These compilers should enable Id to become widely available.

Conclusion

There is an interesting parallel between parallel programming today and sequential programming in the early 1950's. At that time, programs were written in assembly language, and programmers had to be acutely aware of the architectural details of their machines. There was widespread skepticism that high level, machine-independent programming was possible with acceptable efficiency. John Backus and his group changed all that, with their outstanding FORTRAN implementation. Today, parallel programmers have to be acutely aware of the architectures of their parallel machines, and there seems to be some widespread skepticism that high level, machine independent parallel programming is possible with acceptable efficiency. Our hope is that dataflow languages like Id can do for parallel programming what FORTRAN did for sequential programming. The prospects look bright.

References

1 Introduction

Dataflow architectures often seem mysterious compared to more familiar von Neumann computers. This reflects the history of their evolution, which has tended to emphasize tokens flowing on dataflow graphs as the metaphor for their operation. Recent advances in dynamic dataflow architectures, as well as in our understanding of them [4], make it possible to see dataflow machines in a more conventional light. While the tokens-on-a-graph metaphor is still a useful way to reason about the machine, and especially about a particular compilation style for the machine, one can also think of a dataflow program as multiple, interacting sequential threads [6]. This makes a dataflow machine look more like a typical multiple-instruction, multiple-data (MIMD) multiprocessor, and highlights its strengths relative to other MIMD architectures.

This article describes the Monsoon architecture from this point of view. Monsoon [5, 2] is a prototype dataflow computer currently under-construction in a joint effort between the Massachusetts Institute of Technology and Motorola, Inc. Most of the discussion directly applies to any dynamic dataflow computer with an explicit token store, including the Sandia Epsilon-2 [3], and the ETL EM-4 [7].

2 Parallel Machine Language

Dataflow architectures are perhaps the first general purpose computers that execute a parallel machine language. It is a parallel machine language because it has primitive operations for managing parallel computation, namely, fork and join. A fork instruction initiates a new, independent thread of computation in the machine, with its own program counter and register set. This is depicted in Figure 1a; as the fork instruction is similar, except that the response message (this is the second phase), which initiates the new thread in the processor that originally executed the fork. This new thread may begin a totally independent computation, or synchronize with the original thread issuing the fork continues processing, and a new thread is initiated. The new thread will have its value register initialized to the value fetched from L. In the machine, the fork instruction actually operates by sending a special request message to the processor containing L (this is the first phase), and continuing on to the next instruction in the thread. The remote processor sends back a response message (this is the second phase), which initiates the new thread in the processor that originally executed the fork. This new thread may begin a totally independent computation, or synchronize with the original thread through a join instruction. As long as there is a sufficient supply of parallel activity (i.e., other threads previously forked), the processor will not be idle between the issuing of the request and the receipt of the response.

A split-phase store instruction is similar, except that the response from a store carries no value, only an acknowledgment that the store is complete.

3 Split Phase Memory Transactions

Essential to any multiprocessor is the ability to tolerate long memory latency [1]. Dataflow architectures do this through split-phase memory transactions [1]. From the program's point of view, a split-phase fetch instruction to fetch from location L behaves like a fork: the thread issuing the fetch continues processing, and a new thread is initiated. The new thread will have its value register initialized to the value fetched from L. In the machine, the fetch instruction actually operates by sending a special request message to the processor containing L (this is the first phase), and continuing on to the next instruction in the thread. The remote processor sends back a response message (this is the second phase), which initiates the new thread in the processor that originally executed the fetch. This new thread may begin a totally independent computation, or synchronize with the original thread through a join instruction. As long as there is a sufficient supply of parallel activity (i.e., other threads previously forked), the processor will not be idle between the issuing of the request and the receipt of the response.

A split-phase store instruction is similar, except that the response from a store carries no value, only an acknowledgment that the store is complete.

4 Synchronization

Equally essential to any multiprocessor is the ability to rapidly synchronize parallel computations [1]. The join operation already discussed is the primary means of synchronization in dataflow architectures. In "explicit token store" dataflow architectures such as Monsoon, this operation is performed quite efficiently through the addition of a few presence bits to each word of an ordinary data memory. Every join instruction names a word in that memory, called the rendezvous point. Typically, the addressing mode used to name that rendezvous point is a fixed offset relative to the activation frame (see Section 7). Initially, the presence bits on the rendezvous point are empty. When the first of the two joining threads executes the join, the contents of that thread's value register are stored in the data part of the rendezvous point, and the presence bits are changed to present. The first thread then "dies," meaning that the processor goes on to execute some other thread, created by an earlier fork or response to a split phase transaction. Some time later, another thread executes a join which names the same rendezvous point, finding the presence bits set...
to present, that thread continues with the next instruction following the join, and the synchronization is complete. In the case where the join is part of a two-input arithmetic instruction, the value registers of the two joining threads become operands, one of them having been recorded in the rendezvous point.

Notice that the join operation requires no exotic hardware such as associative memories, only a few extra bits. Notice, too, that every word of data memory comes equipped with presence bits, so there is an effectively unlimited namespace for synchronization events, or, putting it another way, an unlimited number of synchronizations may be in progress at once.

5 Synchronizing Memory Operations

The join operation provides one means for computations to synchronize; another is provided by synchronizing memory operations. These exploit split-phase transactions and presence bits to provide a variety of ways to synchronize producers of data structures with their consumers. An example are the split-phase operations of I-fetch and I-store, which synchronize many readers of a location with a single writer. Before an I-store, the presence bits of the location must be empty; a subsequent I-store request will set them to present as well as storing the data. An I-fetch request to a present location behaves like an ordinary fetch. An I-fetch request to an empty location, however, sets the presence bits to deferred and does not generate a response. A subsequent I-store request will store the value and set the presence bits to present, as before, and also will produce the response to the deferred I-fetch. (How the information to generate the responses is maintained is beyond the scope of this paper; see [2].) Thus, the consumer may issue an I-fetch without knowing whether the data has been stored: the response will not be generated until after the corresponding I-store has happened. This sort of usage generally involves two types of synchronization: the memory synchronization as described above, and a join back at the issuing processor to synchronize the response to the I-fetch with the thread that issued the request.

Monsoon provides several forms of synchronizing memory operations, including take and put operations for building locks; different operations are selected by choosing an appropriate fetch like or store like opcode as the first phase. Of course, ordinary imperative fetch and store operations are provided, their second phases of these operations simply ignore the presence bits entirely.

6 Compiler Controlled Thread Granularity

It is often said that parallelism in dataflow architectures is fine-grained. In fact, the compiler has considerable control over the granularity of parallelism exposed to the hardware. Thread granularity can be measured by the number of instructions a thread can execute before it stops or synchronizes with another thread via a join. Now consider compiling an expression like (c+a[i]) + (c+b[i]). Assuming a and b are global arrays, the fetches of a[i] and b[i] will be split-phase transactions. There are two ways the compiler could express the remainder of the computation: (1) it could join the two responses, then multiply each by c and add the products; (2) it could have a separate thread for each response, each of which multiplies, and then those two threads join together and add. The arithmetic in strategy (1) is in one large thread, whereas in (2) it is in three smaller threads.

The traditional dataflow style of compilation uses arithmetic instructions that always join, and so that style is a fine-grain style of compilation. This is a strategy well matched to non-strict functional languages such as Ld, as their semantics limit the size of grains that any compiler could discover [8]. Larger grains might be employed by a compiler for an imperative language. What is interesting is that because fork and join are so efficient, the choice of large grains versus small grains is often determined by addressing modes and register allocation, as opposed to the overhead of synchronizing. As Monsoon's addressing modes are tuned to the dataflow style of compounding, finer grains can in many cases result in superior code, even though there is more joining.

7 Compiler Controlled Task Distribution

A common misconception is that work distribution across processors in a dataflow machine is at the level of individual instructions, i.e., at every fork. In fact, the thread created by fork always executes on the same processor that executed the fork instruction itself. So strictly speaking, fork does not create any parallel activity, only another thread that is queued up until the processor finishes with the current thread. Creating a thread on another processor requires a special kind of fork instruction called send or start, hence, the granularity of work distribution is under compiler control. Typically, a compiler inserts send instructions as part of a procedure calling convention: a procedure is invoked by creating a new activation frame (stack frame) for the call—the frame may be on a remote processor—and then using send to start one new thread on that processor for each argument, where the value register of each new thread is a parameter for the call. Similarly, the called procedure returns a result to the caller by using send to start a thread back on the calling processor, which may then join with the thread that initiated the call in the first place. Task distribution is at the level of procedure calls, even though instruction scheduling is at the finer granularity of threads.

It has been tacitly assumed up to now that local variables of a procedure invocation, as well as the rendezvous points for its joins, are collected together in an activation frame associated with the invocation. (Activation frames are like stack frames in a uniprocessor, except that because of the possibility of parallel procedure calls they form a tree instead of a stack.) This is accomplished through the use of addressing modes that operate relative to a frame pointer that is part of the register set of every thread. One of the operands to the send instruction is a frame pointer for the new thread; for a procedure call, this would be a pointer to the new activation frame created for the call. The actual routing of the new thread to the target processor is accomplished simply by considering the most significant bits of the frame pointer to be a processor ID.

8 Summary

Though dataflow architectures have grown up in a tradition of radical departures from von Neumann computing, they have progressed to the point where they may be seen as an evolutionary step. The current direction in dataflow processor design is to exploit the new-found similarity to conventional architectures, leading to processors that match von Neumann efficiency on sequential code while retaining the benefits of a truly parallel machine language.


3Because Monsoon is pipelined, at any given time there may be up to eight independent threads executing in different stages. The number of threads queued up for eventual execution may be as large as 32,000.
The design and validation of parallel algorithms is a rewarding and satisfying experience once the implementation has been completed and debugged. It is this latter task which can be extremely frustrating when dealing with a general purpose multiple instruction multiple data (MIMD) computer system. Errors in expressing parallel constructs give rise to unpredictable execution behavior, affecting both the resulting answer and the sanity of the programmer.

The notion of a high-level parallel programming language that insulates the programmer from the perils of asynchronous bugs and booby traps has many degrees of freedom in a language implementation (on language, compiler, and target machine level) for a quantitative comparison to be of interest.

We begin our study by examining four routines that exhibit different computational characteristics. These routines and their corresponding characteristics are listed below:

1. Fast Fourier Transforms (FFT)
   - computational parallelism
   - data dependences among the butterfly shuffles
2. Adaptive Quadratures
   - dynamic unrolling of recursively adaptive grid refinements
3. Eigenvalue/vector Solvers
   - application of cyclic rotations (incremental array updates)
4. Stochastic Simulations
   - data accumulation

Details of our implementation and performance analysis will be presented during the session's presentation. For economy, we chose to describe only one of the four routines that have been analyzed at this time.

FFT

A fast Fourier transform (FFT) routine was written in Id by J. Michael Ashley, a former summer student at Los Alamos National Laboratory. The FFT exhibits both divide and conquer and loop parallelism. The relevant Id program segments are:

\[
\text{Def Main_fft Size of V} = \\
\{ C = \{ \text{array} (1, \text{Size of V}) | [i] = \text{Cmplx} (i * 1.0) 0.0 \} | j < 1 \text{ to Size of V} \};
\]

In FFT C:

\[
\text{Def FFT V} = \{ \text{Size of V} \text{ is bounds V} \};
\]

In (if (Size of V == 1) then V else

\[
\{ [\text{Left V}, \text{Right V}] = \text{Shuffle V}; \text{FFT L} = \text{FFT Left V}; \text{FFT R} = \text{FFT Right V}; \text{Mid} = \text{fix (Size of V/2)};
\]

\[
\text{X} = \text{Two Pi/Size of V};
\]

\[
\text{Coeff} = \{ \text{array} (1, \text{Mid}) \}
\]

\[
| [i] = \text{Cmplx} (\text{cos (X * (i-J))} + \text{sin (X * (i-J))}) \}
\]

\[
| j < 1 \text{ to mid }\};
\]

\[
\text{Prod} = \{ \text{array} (1, \text{Mid}) \}
\]

\[
| [i] = \text{Cmplx Mul Coeff(i) FFT R}[i] \}
\]

\[
| i < 1 \text{ to mid }\};
\]

\[
\}
\]

\[
| [\text{array} (1, \text{Size of V}) \}
\]

\[
| [i] = \text{Cmplx Add FFT L}[i] \}
\]

\[
| j < 1 \text{ to mid }\};
\]

\[
| [\text{mid}+i] = \text{Cmplx Sub FFT L}[i] \}
\]

\[
| j < 1 \text{ to mid }\} \};
\]
Shuffle.V = { ( .. Size_of_V ) = bounds V ; Mid = fix ( Size_of_V / 2 ) ;
In ( ! array ( 1 , Mid ) [ i ] = V [ ( i = 2 ) - 1 ] | i < - 1 to Mid ) ;
( array ( 1 , Mid ) [ i ] = V [ ( i ^ 2 ) ] | i < - 1 to Mid ) ) ;

Note that in FFT a recursive invocation of FFT is applied to Left.V and Right.V (the odd and even elements of V, respectively) until the butterfly shuffle on V has been completed. The array element data dependencies occurring in the recombination of smaller results into larger ones are expressed in three array comprehensions defining the values of the arrays Coeff, Prod, and the result of FFT. Running the above program under the /d world simulator for a Size_of.V of 128 gives us the following parallelism profile:

Fig. 1. Parallelism profile of FFT 128.

ANALYSIS

The simulator assumes that each instruction takes one timestep and that the result of an instruction execution is available the next timestep. This approximation leads to an idealized graph interpretation in which maximally parallel execution proceeds along a critical path via a sequence of indivisible timesteps. The graph in figure 1 plots the number of instructions executed at each timestep, and consequently pictures the ideal parallelism profile of FFT 128.

When studying Fig. 1, we observe the following. First there is explosive divide and conquer parallelism (A), followed by (B) a stretch of low parallelism. A second less significant burst of parallelism (C) follows which dies down to an almost sequential tail (D). For larger problems the two sequential stretches (B and D) are observed more and more. The parallelism profile drawn in Fig. 1 is very disappointing since the computational parallelism is known to be very large. To begin to understand this, we know that the FFT program takes O(log(n)) parallel steps to unfold all FFT and Shuffle functions. This accounts for the first burst (A) of the divide and conquer parallelism. Once the functions have been unfolded, the loops (array comprehensions) dictates the parallelism and consequently the speed of the computation. In the first instance the dominant loop is the loop comprehension in the main function creating the ideal parallelism profile of FFT 128.

Since a considerable amount of work may go into the activation of a loop-body, a loop analysis is performed on two simple loops:

Def WW n m = { s = 0 ; r = 0 ;
In { while ( s < n ) do next s = s + 1 ; next r = w m ; finally r } ;
where
Def WW n = { s = 0 ;
In { while ( s < n ) do next s = s + 1 ; finally s } ;

Through an analysis of these two loops, it is found that WW (a doubly nested loop) requires five (5) steps on the critical path to instantiate each inner loop-body, that is, every five parallel steps a new inner loop is spawned off. As the inner loop bodies are skewed on top of each other, the number of them that run in parallel is equal to the critical path required to execute an inner loop divided by the rate at which the loop can spawn off the next inner loop. Figure 2 sketches a parallelism profile for WW 64 20:

Fig. 2. Parallelism Profile for a nested loop.

Clearly, the loop rate plays an important role in the parallelism of a program. Phase B in Fig. 1 show this loop behavior where the dominant loop is the array comprehension in the main function creating the original function values to be transformed. Once every 5 parallel steps an array element is created and sent through the log(n) stages of the FFT.

The completion of the FFT is dependent on the loop rate and the creation of the last two elements of the original array. When the last two elements go into the FFT (phase C in Fig. 1), the remaining stages of the computations can be done with divide and conquer parallelism.

The last sequential tail (phase D figure 1) is caused by the array comprehension in FFT that generates the resulting array.

SOLUTION

The solution to this problem is to unroll loops fast enough so that they don't cause unnecessary delay. This has been recognized by a number of dataflow teams, who invented special instructions (very similar to vector instructions) to rapidly create parallel workloads especially in loops (iterative instructions, repeat mechanism). Id does not have this type of instructions, but it is still possible, although at a considerably higher cost, to create array elements at a higher rate in a nested loop. Take the following array comprehensions:

Def Ab n = { array ( 1 , n ) [ i ] = i | i < - 1 to n };
Def Ab n = { array ( 1 , n ) [ i ] = i | j < - 1 to n by 16 & i < - j to j + 15 };

Some statistics:

<table>
<thead>
<tr>
<th>A</th>
<th>n</th>
<th>S1</th>
<th>Sinf</th>
<th>Ab</th>
<th>n</th>
<th>S1</th>
<th>Sinf</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>223</td>
<td>101</td>
<td></td>
<td>16</td>
<td>277</td>
<td>118</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>399</td>
<td>181</td>
<td></td>
<td>32</td>
<td>504</td>
<td>123</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>751</td>
<td>341</td>
<td></td>
<td>64</td>
<td>958</td>
<td>133</td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>2863</td>
<td>1301</td>
<td></td>
<td>256</td>
<td>5682</td>
<td>198</td>
<td></td>
</tr>
</tbody>
</table>

In the above table, S1 stands for the number of instructions executed and Sinf stands for the total critical pathlength. Where array elements are created at the loop rate (5) in A, they are created at a rate of 16 elements in 5 parallel steps in Ab (i.e., an element production rate of 16/5). This comes at the cost of higher S1 figures. This trick can be employed to make all loops in the FFT program go at a higher element production rate. Applying this idea to Main_fft, we have:

Def Main_fft Size_of_V =
{ C = { array ( 1 , Size_of_V ) [ i ] = CmplX ( i * 1.0 ) } 0.0
  | j < - 1 to size by 16 & i < - j to j + 15 } ;
In FFT C ;

To apply this technique to FFT, requires keeping track of the shrinking vector lengths. The application of this technique to the array Coeff in FFT results in:

745
if Mid > 16 then
  \[ \text{Coeff} = \{ \text{array}(1, \text{Mid}) \} \]
  \[ i \in [1, \text{Mid}] \text{ and } i \in [j, j+15] ; \]
  \[ \text{In } \{ \text{array}(1, \text{Size_of_Y}) \} \]
  \[ [i] = \text{Add_c FFT_L[i]} \text{ Prod}[i] \]
  \[ j \in [1, \text{Mid}] \text{ by } 16 \text{ and } i \in [j, j+15] \} \]
else
  \[ \text{Coeff} = \{ \text{array}(1, \text{Mid}) \} \]
  \[ i \in [1, \text{Mid}] \text{ and } i \in [i-1, i-1] ; \]
  \[ \text{In } \{ \text{array}(1, \text{Size_of_Y}) \} \]
  \[ [i] = \text{Add_c FFT_L[i]} \text{ Prod}[i] \text{ if } i \in [1, \text{Mid}] \} \]

Its parallelism profile is shown in Fig. 3.

\[ \text{Fig. 3. Parallelism of unrolled loops in FFT-128.} \]

The critical pathlength is about 30 percent of that of the original program, and the two ugly sequential stretches B and D have disappeared. Now the parallelism of the program is satisfactory.

It turns out that the array comprehensions are rather inefficiently implemented in Id. In the talk we will address the efficiency issue in some detail.
CONSTRUCTIVE ALGORITHMS AND PRUNING: IMPROVING THE MULTI-LAYER PERCEPTRON

Mike Wynne-Jones
Research Initiative in Pattern Recognition, RSRE, St. Andrews Road, Malvern, WR14 3PS, UK. Phone: (+44/0)684 894344, fax: (+44/0)684 894540
E-mail: mikewj@uk.mod.hermes

Abstract

A number of techniques have emerged recently, which attempt to improve on the multi-layer perceptron training algorithm [1], by changing the network architecture as training proceeds. These techniques include pruning useless or unnecessary nodes or weights, and adding extra nodes as required. The advantages to be gained are smaller networks, faster training times on serial computers, and increased generalization ability, with a consequent immunity to noise. In addition, it is frequently much easier to interpret what the network is doing. One can then begin to draw analogies with other pattern classifying techniques such as decision trees and expert systems. We review these techniques, suggesting the classes of problem to which they are most applicable, and indicate possible future work in this direction.

1 Introduction.

Multi layer-perceptron networks are well established as a standard neural network technique for pattern recognition tasks. They are harmed, however, by a number of problems and arbitrary parameters which make them much less dependable and predictable than they could otherwise be. The main problem is that there is no way of determining in advance how many units there should be in the hidden layer. If there are too few, the network may not learn at all, while too many hidden nodes lead to over-learning of individual samples in the training data, at the expense of forming an optimal model of the data distributions. This leads to an inability to generalise, so that previously unseen data are labelled according to the nearest training sample, rather than in accordance with a good model of the problem. Despite numerous analytical and heuristic attempts to determine this number, [2,3] no general and reliable method has emerged until recently.

Recent developments fall into two categories. Pruning algorithms build and-train a large network, and then remove nodes which contribute little to the network's operation, while constructive algorithms attempt to form an approximate solution using a small network, and then add further nodes to increase the precision as required. It has been demonstrated [1] that a larger network is generally required to learn a classification task, than is needed simply to implement a known solution using predetermined network weights. To this end, an ideal algorithm for determining network architecture might add nodes during early training, and apply pruning selectively as a solution is approached. The need for additional nodes to enable the learning of a solution arises because a network of the minimal size to solve a problem is likely to get stuck in one of many locally optimal solutions in the training phase. In constructive algorithms we hope that the local optima in which we find ourselves will represent a reasonably good approximation to the true solution, and we escape it by the addition of more free parameters to the network. To work well, the new parameters must be added with predetermined values so that the solution of the new network is at least no worse than that of its parent. By pruning weights (or nodes) which do not contribute significantly to the classification or mapping task, we hope to reduce to a minimum the number of degrees of freedom used by the network to implement the solution, thereby ensuring we have a simple model of the system. One way of making this obvious implementation of Occam's Razor, is to minimise the total activation in the hidden layer, which encourages the nodes there to act orthogonally. This is the same behaviour for which we aim in inductive inference, i.e. in decision trees and automatically generated expert systems. It may be possible to extract information directly from the weights of such a network, allowing us to determine the problem-solving rules that have been learned. Naturally the reduction of the number of nodes to a minimum detracts from the fault-tolerance usually associated with neural networks, and so some thought is needed before such minimalist techniques are used for a particular application.

2 The MLP training algorithm.

We briefly summarise the training algorithm described in [1]. In the notation, $o_j$ denotes the output of unit $j$, upon presentation of pattern $p$. Unit $j$ is connected to unit $i$ by a synapse or weight of strength $w_{ij}$, and computes

$$o_j = f(\sum_i w_{ij} o_i)$$

(1)

where $f$ is a differentiable, non-linear function. The desired output for output unit $j$ on presentation of pattern $p$ is denoted $d_j$, and the global error of the network after the presentation of some training patterns is defined as

$$E = \frac{1}{2} \sum_j (o_j - d_j)^2$$

(2)

where the inner sum is over neurons considered to be output nodes, and the outer sum is over the patterns that have been presented.

Optimisation is achieved usually by gradient descent, Newton's method, or conjugate gradients, all of which use the derivatives of the network error with respect to the parameters to be optimised. We define an error measure in terms of the activations of output units, which can be differentiated; the term back-propagation refers to the repeated calculation of the corresponding derivatives for the previous layer. The chain rule for partial derivatives leads to the generalised delta rule[11] which indicates how these quantities and the appropriate weight updates are to be calculated. Minimization of the sum squared error with respect to the free parameters in the network leads the network to approximate the Bayes discriminant vector, the probability of a class given the input to the network. [31]

3 Constructive Algorithms.

Early constructive algorithms built multi-layer feed-forward networks of perceptron units,[7] which could be applied to problems involving binary input patterns. Convergence of such a network is guaranteed if the training data is linearly separable.[8] The Pocket Algorithm[9] is an extension of the Perceptron Learning Rule,[10] which allows approximate solutions to be found for non linearly separable problems, by holding in one's pocket the best set of weights found so far. For binary patterns, it is always possible to cut off a corner of the binary hypercube with a plane, thereby ensuring that we can learn a binary function of binary patterns by repeatedly adding perceptrons to a network. These techniques have been implemented in the Thin [11] although this has been superseded by the Epoint algorithm[12,13] which is more efficient in terms of the number of nodes created. They build tree-like networks with finer resolution of the input space at the leaves of the tree, and do not usually include a stopping criterion to halt the addition of new nodes or layers. This means that every sample in

*Employed by THORN EMI Central Research Laboratories
the training set is learned, but has strong repercussions if the training
data is incomplete, has noise, or is derived from a classification prob-
lem where the class distributions overlap. These networks are good for
learning completely a logical data set, but cannot form a useful model of
many statistical distributions.
Later methods [14,18,34] apply to non-binary functions of non-
binary inputs. They usually build a single hidden layer, which has an
advantage over the 'deep network' methods that the propagation time
for data from the input to the output is shorter, and constant. This is
particularly important for real time signal processing applications,
but not usually in simulated networks used for classification. They do
not guarantee to learn every sample in the training set, but are more
likely than the earlier algorithms to converge to solutions with good
generalization ability for statistical problems.

3.1 Tiling Algorithm and Upstart Algorithm.
The Tiling Algorithm builds a layered network, with each layer ap-
proximating the solution of a binary function of classification problem
more closely than the previous one. A master unit in each layer learns
at least one new pattern, and the ancillary units ensure that no pre-
viously learned patterns are lost. The algorithm creates only as many
nodes as it needs to build a solution, and it has been demonstrated to
have good generalization capabilities. On the down side it requires
an excessively large number of nodes, as most ancillary units just dupli-
cate the action of those in the previous layer. This problem is avoided
in later methods (Upstart, Cascade Correlation), by allowing connec-
tions to span many layers. A recent method which has grown out of
the Tiling Algorithm is the Neural Tree [35] which makes repeated use
of the Pocket Algorithm to divide the input space into partitions for
classification, and builds a decision tree [36] as it does so. No results
have been published to date, but the method looks very promising.

The Upstart Algorithm builds a binary tree of nodes. For each
node, one subtree corrects all errors where a one is expected, and the
other corrects all errors where a zero is expected. Each subnode is
guaranteed to classify at least one of its targets correctly, and so con-
vergence is guaranteed for the problem. The number of nodes grows
linearly with the number of training patterns for the theoretically har-
est problem, a random boolean function, and this is better than the
node growth rate of the Tiling Algorithm. Training can be speeded
up if each subtree is trained using only the patterns that have not
already been learned elsewhere in the tree. Extensions are possible,
whereby the trained network is mapped directly onto a feed forward
network with only one hidden layer. The Upstart Algorithm is eas-
ily extended to classifiers with several possible output classes, while
preserving many of its advantages.

3.2 Dynamic Node Creation in Standard MLPs
Ash [15] describes a system where the error rate is analysed as train-
img proceeds, and a new hidden node is added whenever the error rate
is no longer decreasing significantly, but is not acceptably low. The new
nodes are introduced with small random weights, and the results are
encouraging when compared with the training times required by stan-
dard networks of fixed size. The scheme includes an analysis of when
new nodes should be added, based on the assumption that the error
rate decreases exponentially as training proceeds, but the author feels
that a better way is needed for determining the initial weight values
of the new nodes, as small random weights do not guarantee that the
larger network is at least as good as the old one.

A similar scheme [16,31] adds new nodes to the hidden layer, but
speeds up training by freezing the weights of the previous ones. Back
propagation's property of attempting to force each hidden node to
account individually for all the error in the output layer means that
the newly introduced nodes learn the errors of the previous network
with frozen weights. Interestingly, this feature is usually considered a
disadvantage, as the use of 'teamwork' amongst nodes might be ex-
pected to find a solution more quickly.

3.3 Cascade Correlation
In this algorithm, which applies to real-valued inputs and outputs, hid-
en units are added with inputs from all previous input and hidden
units. [14] The weights to all previous units are frozen, and the new
unit learns a mapping which has the best possible correlation with the
errors of the previous network. The weights to the new unit are then
frozen, and the process continues until there are no more errors. This
method is not limited to binary classifications, although published re-
sults only cover these problems. There is no back-propagation, with
the result that the networks train quickly; the speed compares very
favourably with both backprop and 'quickprop' [19,20] when learning
to separate two interlocked spirals. The problem of learning individual
samples from overlapping distributions might be addressed by con-
inuing the development of a network until the error figure found by
correlation decreases to the value expected from prior knowledge of
the problem. This could be a noise model, or a measure of the degree
to which the data distributions are known to overlap. Since Cascade
Correlation builds tall networks, with many layers and connectivity
from each layer to all earlier layers, it has the advantage of enhanced
feature detection over the networks described in the previous section.
On the other hand, the many-layered architecture leads to a variable
delay between application of an input and the appearance of the result.

3.4 Meiosis Networks.
Meiosis networks [17] arise from an approach to multi-layer percep-
trons which combines the usual gradient descent optimization with a
stochastic search. This has the advantage that it is possible for the
network to escape from a local minimum by a random-perturbation
in the weight space, with a probability which decreases as the net-
work approaches a good solution. This is implemented using stochas-
tic weights, sampled from gaussian distributions each time a weight
value is required. The learning algorithm updates the mean and the
standard deviation of this distribution. Thus there are three learning
rules: the mean is updated according to the normal learning rule de-
scribed in §2. The standard deviation is increased at each update to
reflect the uncertainty indicated by a large update, and decays with
time, allowing the weights eventually to become deterministic.

Meiosis is the process of one node splitting to create two new ones.
The composite variance of input and output weights is computed for
each node in the hidden layer, and the split occurs for any node whose
composite variance, that is the standard deviation relative to the mean,
is greater than 100%.

$$\sum_i \sigma_{ij} > 1 \text{ and } \sum_i \sigma_{ij} > 1.$$  \hspace{1cm} (3)

Each new weight has a mean which is a jittered copy of the original,
each half the variance of the old weight distribution. This kind of
splitting policy has the advantage that it does not converge to
a complete fit of the training data, and consequently the result-
ning networks are likely to exhibit better generalization than the ones
produced by the Pocket, Tiling and Upstart algorithms. It has the ad-
ditional advantage that the decision on whether to split is made using
only locally measured parameters.

3.5 Summary: What to do next with constructive algo-
rithms
The author feels that constructive algorithms lack a good mechanism
for determining the weights of a newly added node, although Meiosis
is good in the context of stochastic weights. Accepting the idea of
adding new nodes by splitting an old one in two, we also require a
good measure of the degree to which a given node requires splitting.
In our own work [21] we propose a new split mechanism, and build on
earlier ideas from pruning, to obtain an ordered list of nodes with the
most probable at one end, and the most splittable at the other.
4 Pruning

Pruning has been carried out on networks in three distinct ways. The first is a heuristic approach, based on identifying which nodes and weights contribute little to the mapping. After these have been removed, additional training leads to a network which is better than the original.[4] An alternative technique is to include terms in the error function, so that the weights tend to zero under certain circumstances. Zero weights can be removed without degrading the performance of the network. Finally, if we define the sensitivity of the global network error to the removal of a weight or node, and evaluate it for each such parameter in the network, we can then remove the weights or nodes to which the global error is least sensitive. The sensitivity measurement does not interfere with training, and involves only a small amount of extra computational effort. It is also well matched for implementation alongside the node-adding scheme described in §3.5.

4.1 Heuristic Pruning

Sietsma and Dow [5] have described a pruning scheme based on analysing automatically the activations of nodes, in response to different patterns in the training data. Their pruning takes place in three different ways:

- Pruning of units that do nothing, or duplicate the action of other units. A unit is redundant if it has the same output for all patterns in the training data, or duplicates the action of another node.
- Pruning of units that provide unnecessary information. If a node makes a distinction between patterns that are later recombined, then the node can be deleted.
- Pruning out an entire network layer. A network with many layers may well have one or more of them redundant, especially after the earlier stages of pruning have been carried out. Indeed, it has been shown [21,33] that one hidden layer is theoretically sufficient to implement any problem, although this may require more nodes than a multi layer network.

4.2 Pruning which is inherent in the learning algorithm

This lower-level approach is more appealing, as the constraints on structure for the network come from the network itself, rather than from a global monitoring system as described in §4.1. On the other hand, systems have been proposed where processes such as adding nodes and pruning would be carried out by exactly this type of global control process, which could itself be a rule based system or a neural network.

4.2.1 Minimising a Biased Cost Function

In standard learning techniques we optimise a network in terms of a cost function which describes the goodness of fit of the model stored in the network, to the data representing the problem to be learned. A second term can be added to the cost function [e.g.,4] incorporating prior knowledge of the problem, such as the kind of architecture which is likely to be most useful for the problem, constraints on weight values, or constraints on the function implemented by the network. A constraint on the function implemented might be that the network should not learn to classify individual samples from overlapping or noisy distributions; elimination of such unlikely solutions leads to the extra term being referred to as a bias or regularizing term in the field of non-parametric statistics.

\[ O = E + B \]  

In relation to pruning networks, or the production of minimal architectures, the constraints we wish to apply by means of the bias term will be such that they minimise the number of active nodes or weights in the network. These can be removed from the network before training is continued.

Rumelhart’s pruning mechanism (unpublished), involves a bias term of

\[ B = \sum_i \sum_j w_{ij}^2 \]  

(5)

giving a modified weight update rule

\[ w_{ij}(n+1) = -c \frac{\partial E}{\partial w_{ij}}(n) + \beta w_{ij}(n) \]  

(6)

This causes the weights to decay exponentially towards zero, and this approach was successfully used by Hinton in some experiments on analysing the use of hidden layers and a representation bottleneck to encourage good generalisation.[32] Weights which come sufficiently close to zero in the weight decay scheme can be pruned out. This work was taken a step further by Hanson and Pratt [22], and Rumelhart, with the aim that high and low weights should decay strongly, while mid-range weights are left unaffected. High weights are discouraged for reasons of smoothness, while low weights can be pruned. These experiments showed that different bias terms are indeed useful for finding minimal architectures, but the bias term could not easily be used in conjunction with the momentum term used traditionally for avoiding local minima. Perhaps the biased cost function could be combined with Hanson’s stochastic search MLP so that the traditional momentum term would no longer be needed.

Biased cost functions have been used in other applications too, first for pruning nodes by including a total node relevance term in the cost function,[28] The relevance term is defined as a product of functions of the weights into and out of a node, and minimising the cost function then minimises the total number of relevant nodes. This kind of node decay (cf. weight decay, above) has not been analysed in detail, but gives promising initial results. A second example encourages the optimal use of hidden units,[29] by including the hidden layer activations in the cost function to be minimised, and hence forcing the units to act orthogonally. This attempt to avoid unnecessary units right from the start of training, rather than allowing them to develop and eliminating them later. This technique could prove useful in the context of integrating neural networks with expert systems or decision trees, since these systems attempt to invoke independent rules wherever possible.

The initial aim of Rumelhart’s use of the biased cost function was to avoid large weights, and consequently to ensure smoothness of the mapping implemented by a network. A sigmoid transfer function is linear for small input values, but approximates to a step function for larger values (and hence for small weights) step functions are necessary to classify individual samples from a statistical distribution and we have already been emphasised in §3 that this should be avoided if at all possible.

An alternative approach to discouraging the fitting of individual samples, again emphasizing the power of the biased cost function, is to include a term in the cost function representing the curvature of the mapping, averaged over the input domain.[30] The elimination of high curvature eliminates sharp transitions, and has a similar effect to the bias term in equation 5. In addition, if the data distributions are smooth, and the curvature of the mapping is minimised, the performance is likely to be improved for the classification of previously unseen patterns. Minimal networks implementing low curvature mappings would also be more likely to perform usefully in extrapolation, that is for the classification of patterns outside the input domain represented by the training data.

4.2.2 Pruning according to an ordered list of node or weight relevances

Mozzer and Smolensky[21] opt for an ‘all or none’ approach to pruning, rather than gradual pruning by means of weight decay. They investigate the relevance of a unit in the network, defined as

\[ r_i = E_{\text{with unit }i} - E_{\text{without unit }i} \]  

(7)

so that the relevance of a unit depends on how much the network global error will increase if the node is removed. Pruning could then be ap
The relevance measure is determined by the shape of the error surface around the minima-to which the network has been trained. This is best characterized for a parabolic minimum by the first and second derivatives according to the Taylor series, namely

\[ E(a) = E(\bar{a}) + \frac{\partial E(a)}{\partial a} |(a - \bar{a})| + \frac{1}{2} \frac{\partial^2 E(a)}{\partial a^2} |(a - \bar{a})|^2 \]  

(8)

although Moez and Smolensky propose to approximate it by carrying out a single pass of back propagation with a linear error function, \( E = |\phi - \phi_t | \), so that the gradient is not zero right down to the error minimum. The question of what parameters \( \phi \) _"should_" have been used as the measure against which the shape of the error surface is measured, is resolved by introducing the _attentional strength_, \( \alpha \) of a unit, where the output of the unit is now \( \phi_j = \sum \omega_j \phi_j \phi_j \alpha \) and the relevance can be expressed as \( \rho_j = E(\phi_j(\alpha) - E(\alpha)) \). Relevance is determined by the shape of the error surface around the value \( \alpha = 1 \). This pruning technique led to networks from which rules could be extracted quite easily for low dimensional boolean problems, and the relevance measures facilitated an evaluation of which rules were the most frequently _"voted_" in a given problem, ie. which splits in the feature space are the rules, and which are the exceptions. LeCun, Denker and co-workers [25,26] have carried out similar experiments for pruning weights, but used the second derivative term from the Taylor series, instead of the modulus approximation. The second derivatives are found by back-propagation [25,26], (or by measurement[27]) and the results are very promising. The number of free parameters in the networks used to implement handwritten digital recognition [28] was reduced by a factor of four. There are no results available of comparisons between this pruning methods and those discussed earlier, but this gives by far the most convincing way of identifying which nodes are to be pruned. The idea of an ordered list of sensitivities of the network global error to each weight was used also by Karnin.[28] Locally the sensitivity would be determined by integration over the entire weight space, but since this is not possible, (cf. training by exhaustive search), it is \( \rho \), _tagged_ just along the training path. It seems likely that the sensitivity found in this way would depend most strongly on the starting weights (ie. random) and not be as good as the measures described above.

5 Discussion

We have discussed a number of techniques which aim to build a neural network whose size, and hence whose number of internal parameters, is optimal for the model of a given problem. While some problems are most efficiently modelled by certain types of network, the multi layer perceptron family are a good general learning tool for a wide range of modelling applications. Since we do not know in advance what size to use, and because we usually need a larger network to learn a mapping than simply to implement a known solution, it is sensible to allow a small network to grow during early training, until a reasonable solution is found, and then to optimise this solution during later training, to give a small, fast and efficient network which is an accurate system model.

There are a variety of pruning mechanisms and constructive algorithms are gradually appearing which can add new nodes at suitable times. The weights for the new nodes can be predetermined to ensure firstly that the new network is no worse than the old one, and secondly that it can be expected to find a better solution with further training. We have seen that a measure of the sensitivity of the network to the presence of weight or node forms a good criterion for determining whether it can be removed, and that at the other end of the scale, if applied to nodes, the same measure can be used as a criterion for splitting a node-in two.

Of the current constructive algorithms, _Upstart_ appears to be the best for binary mappings, while the best for real valued mappings is _Cascade Correlation_ Both of these have the problem of long propagation delays from the network inputs to the outputs, although this is solved for _Upstart_ by transforming the trained network. The best criterion for pruning appears to be the measure of the sensitivity of the network error to the presence of a parameter, and it is hoped that this will very soon be used as the basis of an integrated system for building networks, incorporating construction in early training and pruning as a solution is reached, enabling an optimal architecture to be found.

6 References


16. N Thacker, AIVRU, Sheffield University, UK, personal communication.


18. M Wynne-Jones, 1990. RIPREP/1000/82/90, Research Initiative in Pattern Recognition, RSRE, Malvern, WR4 3PS, UK.


Neural Networks as Petri Nets

Hans Nieters, GMD, 5205 St. Augustin, Postfach 1240, Germany
Email: niets@gmd.de

Abstract

Relative few papers on the relationship between Neural networks (NN) and Petri nets (PN) are known (e.g. [1], [6]). Other groups try different approaches: M.J.Murre (Leiden University, The Netherlands) and K.Lautenbach (University Koblenz-Lindau, Germany). Our approach is to use Predicate Transition systems (PT systems). The result is, that for nonrecurring NN's a behaviorally equivalent PN can be given. The proposed technique of transforming NN's (with and without backpropagation) into PN's is demonstrated by examples. We show some differences between both modeling techniques and also, that both could perhaps gain benefits from each other, if they were put on a common basis.

NN as PrT systems

We follow about the NN definitions given in [4]. Predicate Transition systems (PrT systems) are a widely used high level class of PN. Details about PrT systems can be found in [3], [5]. The representation of a NN (without backprop.) into a PrT system is done by associating to each unit a transition and to each link a place. Weights occur in the 'equations' associated to transitions, where also the computation of activation values is performed. Example: The NN for the XOR-problem is shown in Fig. 1, the corresponding PrT system in Fig. 2.

![Figure 1: A NN for the XOR problem.](image)

The input marking corresponding to the input pair \((x,y)\) is placed in the input places \(i_1, i_2\) such that \(M_0(i_1) = x, M_0(i_2) = y\), where \(M_0\) initial marking. The corresponding PrT system is 'behaviorally equivalent' to the NN, in the sense that for each input given to the input units of the NN, the marking of the output place(s) becomes (after firing of the transitions) identical to the output values of the output units (after updating the corresponding units).

Let's explain the PrT system of Fig. 2 in detail. The transitions correspond one-to-one to the units of the NN (1 to 6). For input units we add input places \((i_1, i_2)\), for output units an output place \((o_5)\). For each link between two units we generate a place and connect that with the units. The arcs are labeled i.e. with formal sums of tuples \(<t_1, t_2, \ldots , t_m, t_m>\), where the \(t_j\) are either variable symbols (starting with capital letters) or terms over some set of operation symbols and variable symbols. In our example \(m = n = 1\) and each \(t_j\) is a variable symbol. The markings are taken from that set of formal sums too, but no variable symbol must occur in the \(t_j\). The labels of arcs in the neighborhood of a place must be of the same arity (it)

Let's explain the PrT system of Fig. 2 in detail. The transitions correspond one-to-one to the units of the NN (1 to 6). For input units we add input places \((i_1, i_2)\), for output units an output place \((o_5)\). For each link between two units we generate a place and connect that with the units. The arcs are labeled i.e. with formal sums of tuples \(<t_1, t_2, \ldots , t_m, t_m>\), where the \(t_j\) are either variable symbols (starting with capital letters) or terms over some set of operation symbols and variable symbols. In our example \(m = n = 1\) and each \(t_j\) is a variable symbol. The markings are taken from that set of formal sums too, but no variable symbol must occur in the \(t_j\). The labels of arcs in the neighborhood of a place must be of the same arity (it).

To each transition a conjunction of formal sums must be given. The proposed technique of transforming NN's (with and without backpropagation) into PN's is demonstrated by examples. We show some differences between both modeling techniques and also, that both could perhaps gain benefits from each other, if they were put on a common basis.

NN with backpropagation

The translation of a NN with backpropagation requires a single 'global' transition for setting the activation values (for input units), the target values (for output units), and for testing whether to stop updating or starting the next update with new input/target vector. We give now the PrT transition(s) for each (input, hidden, output) unit of a NN.

Global takes input from three global places 'init', 'control' and 'trained'. 'init' contains the vector of input and target values for one computation cycle, including -1, +1 values for the threshold input values. For example, in case of the XOR example above, the initial marking of 'init' could be: \([0, -1, 0, -1, -1, 1, 1, 1, -1, 1, -1, -1, -1, 1]\), where \(K\) is the number of input units and all of them are assumed to be 'global' fires. The global place 'trained' initially contains a \(<0, 0>\) token, which is altered to \(<1, 0>\) in case of firing, which may happen only if the equation 'Tr=0' is satisfied. i.e. 'global' fires exactly once for each computation cycle. The transition produces output tokens for each pair of places \(e_j, a_j\), where \(e_j\) runs over the set of input units and \(j\) over the set of successors \(S(i)\) of unit \(i\). The arcs are labeled with a control token \(<0, 0>\), resp. with \(<1, 0>\), where \(i\) denotes a variable symbol (i replaced by the same index as in its corresponding place).
For each input unit \( r \) in \( 1, \ldots, K \) a transition \( u_r \) is generated, which is part of the backward pass (cf. right hand side of Fig. 3). It takes its own activation value \( A_r \) (set by 'global' and used by its successor units), the old weights \( W_{pr} \) (connectors to its successors) and the error output \( D_r \) backpropagated from its successors \( (5) \). It produces new weights \( W_{pr} \) changed by the transition equations, and a token on the 'control' place. The transition has an equation for each outgoing connection. All of them must be computed in order to produce the \( W_{pr} \) values. (Symbols starting with capital letters denote variables), 'alpha' is a constant (0.5 for our example).

For each hidden unit two transitions must be generated (cf. Fig. 4), \( u_{1} \), for the forward pass, computing the new activation value, and \( u_{2} \), for the backward pass, computing the error signal \( D_r \) to its predecessors \( P(r) \) and then the new weights. The equation in \( u_{1} \) uses \( \gamma \), which is a sigmoidal squashing function. In \( u_{2} \), the denotation \( j = 1, \ldots, \#(r) \) is used both as summation index in the first equation (the same as in \( u_{1} \)) and as index for producing the rest of equations.

\[ \begin{align*}
\text{transition} & \text{ for generating the transitions for output units should be clear now. The initial value of 'trained' } \theta \text{ is } < 0 > \text{ and gets changed only if the stop criterion for that transition is satisfied. The weight places } W_{pr} \text{ are initially marked with the initial values. The result of generating the Petri system for the XOR example is left as an exercise to the interested reader.}
\end{align*} \]

**Figure 4:** The PrT transitions for a hidden unit with backpropagation.

**Figure 5:** The PrT transitions for an output unit with backpropagation.

The schema for generating the transitions for output units should be clear now. The initial value of "trained" \( \theta \) is \(< 0 \geq 0 \) and gets changed only if the stop criterion for that transition is satisfied. The weight places \( W_{pr} \) are initially marked with the initial values. The result of generating the PrT system for the XOR example is left as an exercise to the interested reader.

**PN + NN = Hybrid systems?**

Some differences between PN and NN are: (1) Single type of nodes (NN) versus bipartite graph (PN). (2) Continuous activation values (PN) versus discrete (NN). (3) Also negative real values of weights (NN) versus positive integer (PN). (4) Units are often updated synchronously (PN) but asynchronously (NN) (5) NN are deterministic, i.e. no decisions between alternatives are taken even in recurrent NN (not considered here). Hence the PN opens up the possibility to model alternatives additionally. Also the embedding of propositional and first order predicate logic in PN theory has no equivalent in NN. The NN system has a very complex logical system. In PN we know that a propositional implication \( a \rightarrow b \rightarrow c \rightarrow d \) is represented as a denoted transition with \( a \), \( b \) as input and \( c \), \( d \) as output places. The representation of (even) propositional logic in NN is much more complicated. (7) PN are suited for formal analysis (reachability trees and invariant calculus), since PN are used to describe exact solutions for discrete systems, where the main problem is to guarantee that the system is live and safe. Since NN are not used for designing systems like operating systems or protocols, NN units keep only a single activation value and distribute only that single value to all successor units. A restricted form of liveness condition occurs as initial condition in recurrent NN only. (8) Modifications of weights and hence 'learning' is usually not at all considered in PN. Exceptions are self-modifying nets. The usual tools designed for editing and simulating PrT systems could in principle be used for the PN model of NN, but the mere amount of connections up to \( \times 10^3 \) makes it necessary to use dedicated simulators instead of the general simulators for PrT systems at hand. E.g., the GRASPIN simulator (71), applied to the XOR-network with backpropagation, updates the network in 3 seconds, whereas a dedicated NN simulator manages at most 5 to 20 thousand updates per second.

**References**


[2] H. E. Fuss, AFMG - ein asynchroner Flussmodellgenerator. (in German) GMD Bericht Nr 100, 1975


The symmetric logarithmoid: an activation function for neurons

Abhay Bulsari, Alexander Medvedev, Björn Saxén and Henrik Saxén
Kemisk-tekniska fakulteten
Abo Akademi, SF 20500 Turku / Abo, Finland
E-mail: vt.ai@abo.fi

ABSTRACT. The symmetric logarithmoid provides a viable alternative to the sigmoid, while preserving many characteristics of the sigmoid. The sigmoid is very flat when the absolute value of its argument, \(|a| > 10\). In other words, its derivative is extremely small, and has poor sensitivity to its argument. This is the root cause of the very slow rates of convergence during the training phases of neural networks, and relative insensitivity of the network to a fairly wide range of weights. The symmetric logarithmoid overcomes these limitations, despite perhaps creating some others of its own.

Feed-forward neural networks can be used as simulators trained from the gross observed behaviour of a system. This paper illustrates the applicability of the symmetric logarithmoid activation function in a feed-forward neural network for such training exemplified by a system identification problem of pressure drop in a rough pipe. The inputs to the network include viscosity, density of the liquid, diameter and roughness of the pipe, and the velocity. The outputs are the friction factor and the pressure drop per unit length. The networks were trained using the Levenberg-Marquardt method.

The symmetric logarithmoid is continuous, first order differentiable and a simple, monotonically increasing algebraic function. While minimising the error square sum for the outputs, convergence is generally fast compared to the sigmoidal activation function. Extremely large weights are not commonly generated by the training process, but is a usual feature with the sigmoids. The symmetric logarithmoid evidently, does not mix well with other activation functions, especially the sigmoids.

1. Introduction

A lot of research has been done on feed-forward neural networks taking the sigmoid for granted. The sigmoid, however, has its limitations and its applicability is not universal. Sigmoids are meant for outputs constrained between 0 and 1, or -1 and 1. A linear mapping can extend this range. But many variables do not have such limits, and it is not always desirable to map them to a range of 0 to 1 since sensitivity can be lost in the process of mapping. Feed forward neural networks can be used for system identification of processes [1], and one often comes across variables like temperature, pressure, viscosity, concentration, etc. which have no upper limits, although for a system under consideration, they may stay in a particular range. If the range is more than a couple of orders of magnitude, it is customary to deal with their logarithms in the mathematical models.

The sigmoid is very flat when the absolute value of its argument \(|a| > 10\). In other words, its derivative is extremely small, and has poor sensitivity to its argument. This is the root cause of the very slow rates of convergence during the training phases of neural networks, and relative insensitivity of the network to a fairly wide range of weights.

The symmetric logarithmoid, given by the following equation overcomes these limitations, despite perhaps creating some others of its own.

\[ x_i = \frac{a_i}{|a|} \ln (1 + \beta|a|) \]

The sigmoid and the symmetric logarithmoid can be considered to be in a continuum of activation functions. One extreme of the activation functions is the switch (the sign function), a network based on which cannot be trained by any of the optimisation methods meant for continuous functions. The sigmoid alleviates this problem by smoothing the switch near its discontinuity. The symmetric logarithmoid is continuous and first order differentiable. It is a monotonically increasing function with maximum sensitivity near zero and monotonically decreasing sensitivity away from zero, as with the sigmoid. However, the symmetric logarithmoid never becomes insensitive to the argument, and its output is not limited to between -1 and 1. Networks using this function are a bit easier to train, and the convergence is better. The other extreme of activation functions is the linear (identity) function, which finds limited use in our work for statistical purposes. This function, obviously poses no problems to the usual optimisation methods, and the Levenberg-Marquardt method, which we use, converges in a couple of iterations.

2. The flow in a rough pipe

To study the applicability of the symmetric logarithmoid, we considered a simple system of a flowing liquid in a rough pipe, under turbulent flow. The pressure drop in a pipe is of vital interest while sizing pumps or compressors, and for calculating flow rates. It is estimated by empirical correlations such as the one shown below [2].

\[ \frac{\Delta p}{L} = \frac{\rho \mu u^2}{2d} \]

\[ \zeta = [-2 \log \left( \frac{k/d}{3.7} + \frac{2.51}{Re^{1/2}} \right)]^2 \quad Re > 2300 \]

\[ Re = \frac{\rho w}{\mu} \]

The pressure drop, \( \Delta p \) per unit length is proportional to the friction factor, \( \zeta \), a measure of retardation of the flow by liquid viscosity and pipe wall roughness. The pressure drop, obviously, has no natural limits on the positive side and so does the friction factor.

The pressure drop depends on the density, \( \rho \) and the kinematic viscosity, \( \nu \) of the liquid, the flow velocity, \( w \), and the diameter, \( d \) and roughness factor \( k \) of the pipe.

For the sake of training a feed-forward neural network, examples were generated at random in the following range of inputs.

| \( \mu \) | \( 0.5 - 1.5 \times 10^3 \) kg/m\(^3\) |
| \( k \) | \( 0.1 - 1.0 \times 10^{-3} \) m |
| \( \nu \) | \( 0.5 - 5.0 \times 10^{-6} \) m/s |
| \( w \) | \( 1.0 - 10.0 \) m/sec |
| \( d \) | \( 0.1 - 0.5 \) m |

This guarantees that \( Re > 2300 \).
3. The Levenberg–Marquardt method

The Levenberg–Marquardt method [3-6] was used to calculate the weights in the neural networks which minimised the sum of squares of errors. Most algorithms for least-squares optimisation problems use either steepest descent or Taylor-series models. The Levenberg–Marquardt method is a restricted step method, which uses an interpolation between the approaches based on the maximum neighbourhood (a "trust region") in which the truncated Taylor series gives an adequate representation of the non-linear model. The method has been found to be advantageous compared to other methods which use only one of the two approaches.

4. Results

The results give a clear indication of the applicability of the symmetric logarithmoid activation function.

4.1. Training the neural networks

A table of 200 training instances was created by random selection of inputs in the ranges stated in section 2. The pressure-drop per unit length was tabulated in units of $10^3$ Pa/m. Yet, it often had large values (greater than 10) which would require the argument of the symmetric logarithmoid to be very large. Therefore, the logarithm of the pressure drop per unit length (in units of $10^3$ Pa/m) was tabulated instead. Similar results were obtained with actual pressure drop values.

With one hidden layer, the number of hidden nodes was varied between 2 and 7. The configurations with 5 or more hidden nodes resulted in good fits of the training data. The error square sums (SSQ) for (5,5,2), (5,6,2) and (5,7,2) were 0.189, 0.132 and 0.080 respectively. Networks with two hidden layers are not as easy to train as the ones with one hidden layer. This has been observed with the sigmoids, and was observed with this activation function also. The configuration (5,2,2,1) had a SSQ of 0.341, but (5,3,3,1) with 34 weights had a SSQ of 0.0583, less than 0.084, the SSQ of (5,5,1) with 36 weights. The SSQ for (5,4,4,1) was 0.0316.

4.2. Testing the trained neural networks

The trained neural networks were then tested with various velocities, while keeping other variables constant ($\rho = 10^3$ kg/m$^3$, $k = 10^{-4}$ m, $\nu = 10^{-6}$ m$^2$/sec, $d = 0.2$ m). A plot of the logarithm of pressure drop per unit length ($10^3$ Pa/m) vs velocity is shown in Fig. 1.

It can be seen that for various configurations shown in the legend of the figure, the predicted values are quite close to the analytical values. This accuracy is sufficient for engineering purposes. Typical error is about 0.05, which is about 2.5% of the range on the vertical axis.

5. Conclusions

The feed-forward networks trained with the symmetric logarithmoid activation function performed well in the testing phase. Convergence during training was faster than is usually encountered with sigmoids. The weights generated after training were never very large, although that happens often with sigmoids.

The symmetric logarithmoid, thus provides a feasible activation function for the neurons, instead of the sigmoid, when the outputs are not in a well-defined limited range.

References

Robot Vision Based on Coarsely-Grained Multi-Processor Systems

Volker Graefe

Institut für Meßtechnik
Universität der Bundeswehr München
8014 Neubiberg, Germany

Abstract

Concepts are introduced which allow robot vision systems to be designed according to the inherent structure of the task of vision. Practical results obtained with such systems are presented.

Introduction

Coarsely-grained multi-processor systems have been demonstrated in numerous practical applications to be particularly well suited for building powerful robot vision systems <Dickmanns, Graefe 88a, b>. The real-time vision system, BVV 1, is an early example. Conceived in the late seventies, it uses only a few 8-bit microprocessors (Intel 8085A) with speeds of about 50 km/h on an unmarked road <Regensburger, Graefe 90; Solder, Graefe 90>. Its successor, the BVV 2, was conceived in the early eighties <Graefe 84>. Based on much stronger 16-bit processors, Intel 8086, it enabled, for instance, the experimental vehicle VaMoRs in 1986 to follow a road at a speed of 96 km/h, making it by far not only the world's fastest fully autonomous road vehicle, but also exclusively the only one whose speed was limited by its engine and not by its vision system <Zapp 88>. The BVV 2 is an open system based on the standard Multibus I and using commercially available single board computers. This made it easy to replace its parallel processors by more powerful ones later. Such an improved version of the BVV 2 has been used for detecting and classifying obstacles on the road while approaching them with a speed of about 50 km/h on an unmarked road <Regensburger, Graefe 90; Solder, Graefe 90>.

Recently, the BVV 3 has become operational <Graefe 90>. It employs Intel microprocessors 80286 or 80386, augmented by a custom-designed coprocessor for feature extraction. The BVV 3 is intended to be used, for instance, in future experiments with an autonomous vehicle participating in ordinary highway traffic <Graefe, Kuhnert 88>. The BVV 3 should generally perform feature extraction about 5 to 10 times faster than the BVV 2, depending on the task. Tests have indicated that in some vision related applications it is actually more than 100 times faster than its predecessor.

It should be noted that the good performance of the BVV systems is not a result of utilizing any particularly fast electronic components, but rather of a system architecture that matches the inherent structure of the task of robot vision.

System Architecture

A robot's environment contains a limited number of physical objects that are in some respect relevant for the operation of the robot. Among them could be landmarks, obstacles, the pathway, objects to be grasped, or various other objects. At any given moment only selected ones of these objects need to be monitored simultaneously, provided the robot is able to switch its attention from one object to another one within a fraction of a second.

A robot vision system should, therefore, have the ability to observe a small number of objects, say half a dozen, simultaneously. At the same time it should be able to maintain internal models of a slightly larger number of objects existing in the robot's environment but not demanding immediate attention according to the perceived situation.

If the structure of the task to be handled is reflected in the structure of the vision system a particularly high degree of efficiency may be expected. This has been discussed in greater detail in <Graefe 89>. In short, one "object processor" each should exist within the vision system for each relevant object in the robot's environment. Figure 1 shows the conceptual structure. It comprises a video section, a number of object processors, and a situation processor.

Figure 1

Conceptual structure of object-oriented vision systems
Video Section

The cameras, digitizers, and a means for distributing the video data to the object processors make up the video section. In addition, it may contain camera control devices like platforms or lens controllers. Each object processor has direct access to the digital video data. For the efficiency of the vision system it is of utmost importance that all object processors have independent access to these data. Therefore, each object processor should have its own image memory. A single shared image memory is, of course, cheaper, but it tends to create a severe bottleneck.

Object Processors

Each object processor receives digital image data from one or more cameras and outputs a description of one particular external object. The description may relate to the shape, location, state of motion, or other characteristics of the object. An object processor is a conceptual entity, not a physical one. In principle, it does not matter if such an object processor is implemented on exactly one piece of hardware, or if it contains several computing elements, or if several object processors share one computing element. Given the present state of microprocessor technology, an object processor for a typical robot vision application may be implemented on one to three microprocessors.

Situation Processor

The object descriptions are fed into a situation processor. This is, again, a conceptual entity. It will typically be implemented on a number of microprocessors. The main task of the situation processor is to deal with the interactions of the external objects with the robot and among each other. Besides, it assigns objects to the object processors. The situation processor also interfaces with the human operator, exchanges sensor and control data with the robot, and controls the video section, if necessary. The complexity of the situation processor’s task depends highly on the complexity of the robot's task and on the environment the robot is supposed to operate in. Following an empty road, for example, involves hardly a situation to speak of, because the road is the only external object. On the other hand, driving autonomously in ordinary road traffic will certainly require a very powerful situation processor.

Implementation

The vision system BVV 2 has been designed to support structures as shown in Figure 1. Its video section contains four independent video channels allowing the signals from up to four cameras to be processed simultaneously. Each object processor typically comprises two parallel processors, one for a 2-D object model and the associated feature extraction, and one for a spatial temporal object model. The situation processor may then be implemented on separate hardware, possibly a transputer network.

Two new implementations of the object oriented architecture are currently being developed. Both of them will be based on a standard PC equipped with a commercially available video digitizer and a few custom built microcomputers to be utilized for feature extraction and 2-D object models. In one implementation the Intel 376 microprocessor will be used with an expected performance similar to the BVV 2 level and its software largely compatible with the BVV 2. The other new implementation will be based on the Intel 960. Its performance is hard to predict, hopefully it will approach the level of the BVV 3 at a lower cost and smaller size.

References


Acknowledgements

Part of the work reported has been performed with support from the Ministry of Research and Technology (BMFT) and from the German automobile industry within the EUREKA project PROMETHEUS and other projects.
Markov Random Field Models and Parallel Algorithms for 2D Motion Analysis

Fabrice HEITZ, Etienne MEMIN, Patrick BOUTHEMY
IRISA/INRIA, Campus Universitaire de Beaulieu
35042 Rennes Cedex, France

Abstract

The use of Markov Random Field (MRF) statistical models has recently brought new powerful solutions to classical image analysis problems. In recent papers, we have presented a new class of spatio-temporal MRF models which have been applied with success to different basic tasks in visual motion analysis. In this paper, parallelization methods for those relaxation algorithms are investigated and a new hierarchical approach, based on the interactions between different relaxation processes running in parallel at different scales is presented. The hierarchical method has been simulated in the case of optical flow estimation. It shows good performances (quality of the estimates, gain in number of iterations) when compared to sequential algorithms.

1 Introduction

Markov Random Fields (MRF) models have been successfully introduced in several important issues of still image-analysis, such as image restoration, segmentation or edge detection. [6] Our group has recently extended the MRF models to the analysis of image sequences for motion detection, [3], optical flow estimation, [8] and motion-based segmentation, [5]. In visual motion analysis, MRF appear as an efficient and powerful tool for combing spatial and temporal information. For details concerning the models designed in each case, we refer to [3, 8, 5].

The subject of the paper is the study of parallel approaches for the relaxation algorithms, [6] associated to MRF in image analysis. Indeed, one very attractive property of MRF is that, though the models are global and non linear, the involved computations remain local and are intrinsically parallel. We present here an approach based on relaxation algorithms running in parallel at different scales and interacting periodically. This parallel algorithm is compared to sequential stochastic or deterministic algorithms in the case of optical flow estimation. It exhibits fast convergence properties and only requires a small number of processors.

2 Markov Random Fields for motion analysis

The modeling and analysis of images by MRF's has been discussed extensively in the literature. [6]. To extract labels describing motion from image sequences, observations related to the spatio-temporal variations in the image sequence are combined with a priori generic knowledge on the expected solution, in order to derive estimates of the unknown labels. The labels are binary features in motion detection, vectors in optical flow estimation and region numbers in motion-based segmentation, [3, 8, 5]. MRF models describe the local statistical interactions between these different variables. When a maximum a posteriori (MAP) estimate of the unknown label variables is looked for, MRF-based image analysis reduces to the minimization of a global energy function E which depends on the whole observation and label field, [6]. Minimizing the global energy function E is an intricate problem: the number of possible label configurations is generally very large and the global energy function E may exhibit numerous local minima. Computationally demanding stochastic relaxation algorithms are therefore generally necessary to compute exact MAP solutions. Deterministic descent algorithms such as ICs, [2] can often be used instead, when a good initial guess is available.

A simple model for optical flow measurement

We consider here a very simple model for optical flow computation, which will be used for comparison purpose between the parallel and sequential versions of stochastic relaxation (for more sophisticated models including discontinuities and occlusion processing see [8]).

Let f(x,y,t) denote the observed intensity function, where x = (x,y), s ∈ S designate the 2D spatial image coordinates and t the time axis. The velocity vector at point s is denoted ∇f(u,v), s = (x,y,t)

3 Parallel algorithms for stochastic relaxation

Parallel implementations of stochastic relaxation have been considered for global optimization in applications such as computer-aided circuit design, [1] or image processing, [6, 9, 10]. Until now, three main approaches have been investigated: parallelized Markov chains [1, 7] for sequential updating of the image sites, [6, 9, 10], and parallelization of the local label updating, [1].

Parallelized Markov chains have been proposed by Aarts et al for global optimization problems based on simulated annealing and the Metropolis algorithm, [1]. The basic principle of the approach is to run in parallel several relaxation algorithms, each of them exploring differently the space of all possible label configurations and interacting from time to time. This class of algorithm may be run on a SIMD or MIMD machine by assigning each available processor to a different relaxation algorithm. The interaction is based on periodic transfer of global label configurations between the different processors. A similar method has recently been studied by Graffigne, [7].

Our own approach of the problem is also based on parallelized Markov Chains but derives profit from a hierarchical representation.
of data and labels, which leads to an efficient parallel annealing algorithm with a speed-up larger than the number of processors. The method can be outlined as follows: a data pyramid is first constructed by low pass filtering and subsampling the images of the sequence. Accordingly, a hierarchical representation is also considered for the label fields which are estimated in parallel at different resolution levels on reduced grids (Fig. 1). Relaxations are performed in parallel by assigning one processor at each level of the pyramid.

Following [7], the different relaxations are performed at fixed temperature. To the low resolution levels of the hierarchy are associated high temperatures. At low resolutions, coarse estimates of the label field are visited. Since the total number of possible label configurations is reduced at those scales, the coarse configuration space can be visited very efficiently. A high temperature enables to escape from local minima of the energy function. To the intermediate resolution levels are associated lower temperatures. At these levels, the relaxation process becomes more sensitive to local minima and visits the large or medium scale valleys of the energy landscape, [7]. At the finest resolution level a temperature close to 0 is adopted. Very low temperature in the stochastic relaxation corresponds to nearly deterministic descent of the energy function: the estimation of the label configuration is ultimately fine.

The explorations at the different levels are cooperative: every p iterations - one iteration corresponding to a full sweep on the image - a processor attempts to transfer a small label block to the next finer level (Fig. 1). The interaction process is controlled as follows: a block \( B_r^r \) at resolution level \( r \) is interpolated at level \( r-1 \) using simple re-indexation of the label estimates at the missing positions of level \( r-1 \) (Fig. 1). The energy of the resulting local configuration is computed and compared to the energy of the corresponding block \( B_{r-1}^{r-1} \) at the finer resolution \( r-1 \). The local label configuration on the block \( B_{r-1}^{r-1} \) is replaced by the interpolated configuration of block \( B_r^r \) if the latter is better than the former, that is, if its energy is lower. Local energy on blocks are obtained at reduced additional cost from the local characteristics of the Gibbs distribution.

Experimental results

Experiments have been carried out, in the case of the discrete optical flow measurement model, on several synthetic and real world sequences. Three algorithms have been simulated: a sequential stochastic relaxation (SSR) based on the Gibbs sampler with an exponential schedule \( (\lambda = 0.97, \beta = 300) \), a sequential deterministic relaxation (SDR) algorithm known as ICM, [2], and the proposed parallel hierarchical relaxation (PHIR). In our experiments the same parameter value \( \alpha \) (equ. 1) was used for the potential functions at each level of the label pyramid \( (\alpha = 20) \). In every case the convergence criterion was the same: the relaxation was stopped as soon as the maximum number of modified labels between two successive image sweeps (at the finest resolution level) went below some specified threshold (typically 10).

Figure 2 shows the energy plots obtained for a divergent motion on (A = 1.62) images by the different algorithms. Four resolution levels are used in the pyramid and \( n_{\text{max}} = n_{\text{max}} = 8 \) in this case. PHIR produces estimates close in quality to the one obtained by SDR (the energies of the final configurations are respectively 94700 and 57300). The slight degradation of PHIR with respect to SDR originates from the block effect due to the inter-scale interaction model. PHIR exhibits fast convergence properties similar to SDR, but with estimates of significantly better quality. On an average, the computational gain (in number of iterations) of PHIR over SDR, over several sequences, was about 10. An implementation of the proposed algorithm on an IPSC/2 supercomputer is currently investigated.

REFERENCES

PARALLEL COMPUTER ARCHITECTURES USING HETEROGENEOUS PARALLEL PROCESSING STRUCTURES FOR REAL-TIME VISION BASED ON-LINE INSPECTION OF MANUFACTURED PARTS

E. HIRSCH, Ph. PAILOU
Laboratoire des Sciences de l'Image et de la Télédétection, LSIT, Ecole Nationale Supérieure de Physique de Strasbourg, ENSPS, Université Louis Pasteur, 7, rue de l'Université, F-67000 Strasbourg, France.

C. MÜLLER
Institut für Algorithmen und Kognitive Systeme, IAKS, Fakultät für Informatik, Universität Karlsruhe (TH), Fasanengarten, D-7500 Karlsruhe 1, FRG.

U. LÜBBERT, V. GENGEBACH
Fraunhofer Institut für Informations- und Datenverarbeitung, IITB, Fraunhofer Str. 1, D-7500 Karlsruhe 1, FRG

Abstract.
In this paper, we present concepts related to the design of parallel computer architectures for use in real-time vision applications and based on combinations of heterogeneous processing structures. The corresponding realizations will be described and the use of such systems illustrated by a representative application, e.g. the on-line inspection of manufactured parts.

1 Introduction
Most of the current industrial applications of machine vision, relying on image processing systems with high computation power, are concerned with inspection, quality control, assembly, control of manufacturing processes, autonomous vehicle guidance and robotics. Optical sensing of the application environment, evaluation of the images of the scene and physical reaction after interpretation of the image contents are among the most effective and efficient means for the analysis of the environment and for acting in an appropriate way. Machine vision is thus a way to automate the applications previously described in a flexible and intelligent manner. The goal of computer or machine vision is to extract high-level information about the environment from the low-level information contained in one or a sequence of images of that environment. This should contribute to the design and development of so-called intelligent machines. Because of the amount of information contained in an image, machine vision calls for high computing power. Provided the required hardware is available, this leads to the development of visual systems with the aim to solve these problems automatically, with a computation speed compatible with the application to be solved [1], [2].

This naturally leads to the use of parallel computer architectures, and more precisely to heterogeneous pipelined structures. However, the different systems, proposed for industrial applications quoted above, are often not able to provide the performance and processing speed needed at reasonable costs. This is also the case for more sophisticated applications such as, for example, the automated vision based on-line inspection of manufactured parts in order to detect and locate different defects on the part under control for the optimization of production rate. Specific requirements for the vision systems running such vision tasks have in fact to be defined and secondly to be satisfied.

With the technology available today, in order to run the application in "real time", it is necessary to use heavily parallelism for the image processing and some higher level treatments required by the application. More specifically, one has to match hardware to the envisioned processing. This leads to the design and implementation of heterogeneous pipelined structures in which each component is optimized with respect to the type of processing it has to perform. Furthermore, the elements of the pipeline often imply an internal parallelism (for example, systolic processors, arrays of processors such as Transputer networks). Last but not least, the synchronized use of such systems calls for a carefully designed programming environment, relieving partly the user of the burden of programming such systems.

The following section introduces the definitions and concepts involved by the design and implementation of such machine vision systems. Section 3 describes the image processing systems already realized, using two different approaches. The first system, designed and built at the Fhg-IITB Institute, uses a modular functional approach, whereas the second one aims to implement a reconfigurable, flexible and fully programmable structure. Both systems are able to carry out most of the low-level image processing in real-time.

Development of methods and design of new architectures call thus for careful studies and possibly experimentation. This should be facilitated by the set-up of development centers grouping hardware and software resources. A possible configuration of such an integrated development system for computer vision applications will be indicated.

The paper is concluded by an illustration of the use of such systems, an automated system as a solution to the 100 % control of manufactured parts in a FMS environment. The technique used is based on comparison between images acquired through a vision system and the corresponding data gained from a CAD system. Comparison takes places as well at feature level than at image level. All kind of inspections, ranging from conformity checking up to metrology, can be achieved through use of an user friendly planning system.

2 Machine Vision Systems
In vision applications, two different processing steps can usually be defined. The first step deals with the so-called low-level image processing, which transforms an input image into a modified output image in order to enhance the quality of the input image (e.g. noise reduction and distortion correction) and prepare a subsequent feature extraction. Then some segmentation of the enhanced image is done in order to delimit different regions in the image. The extracted features represent (or model) some characteristic properties of the information content of the scene (perimeter, connectivity, moments...). Thus, after having computed a description of the image content, the second step exploits this description in order to compute statements about the meaning of the image content. These statements are then further used for firstly defining and secondly carrying out the actions to be performed on the environment implied by the application. Thus, this interpretation of the images is used to achieve a well defined goal (e.g. recognizing objects, taking decisions, vehicle guidance, inspection, acting on the real-world).

Imaging devices (generally a CCD camera) produce as input for the machine vision system a pixel image. The purpose of the vision system is to transform the low-level image information content of the input image into a number (as small as convenient) of high-level information pieces, in order to deliver a description of the real-world scene. This description can be further used to take a decision, which, in turn, leads to one or more actions to be carried out on the real-world through a feedback loop. Instead of a feedback free processing model, a more complex processing scheme, including a feedback loop called interpretation loop, can be imagined.

Figure 1 above shows the model first proposed by Kanade [3], [4] and modified by Nagel [5], [6], [7]. It is important to see that the model of figure 2 allows iterative image interpretation and eventual control of the data acquisition parameters. Furthermore, the model implies different types of knowledge for the interpretation process in both the image (low level) and scene (high level) domains. More and more industrial vision applications rely on such a processing scheme, as the examples in section 4 will illustrate.
Image processing thus implies very different processing tasks with respect to applications. These image processing tasks are very different in nature and imply very different data structures for the computations. From the system designer point of view, these levels are usually referred as low, medium and high-level image processing (see fig. 2). This sub-dividing of a complete vision task into three processing levels, reflects the nature of the operations to be carried out and the nature of the data structures used. Figure 2 defines the three processing levels and indicates how the different stages are combined in a sequential processing scheme, implementing the direct processing path of the model of figure 1:

- Low level image processing transforms images into images. They are essentially pixel oriented operations: threshold, low-pass filtering, mathematical morphology.
- Medium level image processing needs images as input and gives features as output. Extraction of lists of contours is a widely known example.
- High level operations transform features into features. Other sources of information than the initial images can be used (a priori knowledge, databases, artificial intelligence). High-level processing also has in charge the control of the interfaces between the machine vision system and its environment (e.g., man-machine interfaces, communication network interfaces, actuator interfaces).

Each processing level has to be implemented on hardware specifically tailored for it, in order to perform efficiently its tasks. Taking this matching into account leads then directly to the design of heterogeneous pipelined image processing structures for complex vision applications. The amount of data to be computed is decreasing from the low-level to the high-level stage, and less and less computing power is needed, but more and more flexibility is required.

According to figure 2, the basic structure of an image processing system can also be defined around three parts, having each different requirements to the processing architecture:

1. The acquisition, pre-processing and restitution of images.
2. The processing in the image domain or the so-called iconic processing step. In this case, the needed algorithms and data structures are nearly directly related to the pixel organisation in the image.
3. The model based exploitation of image primitives and content or the so-called symbolic processing step. In this case, the data structures implied are almost exclusively a function of the type of computation to be carried out.

Three basic structures (for a discussion, see for example, [2], [9]), for so-called general purpose systems, are today clearly emerging:

1. Arrays of mesh-connected processors with more or less computation power.
2. Bus and cascaded structures, chaining different hardware modules with more or less flexibility,
3. Bus oriented processing structures with powerfull, more or less specialized micro-processors.

For special purpose processing, systolic arrays or dedicated LSI or VLSI integrated circuits can also be very efficiently used [10]. However, bottlenecks are usually encountered when the low-level pixel data has to be matched with the high-level data structures [11], [12]. It seems therefore not possible to design a system able to perform well for all the algorithms needed in computer vision. However, heterogeneous structures for the three levels of processing, with an integrated control strategy and incorporating different modes of parallelism, appear to be today the best compromise. With respect to the use of such systems, two approaches, exemplified in the next sections, are possible:

1. The first is based on a modular functional approach, where the user has just to choose the appropriate chaining and parameters for image processing operations.
2. The second is based on a fully programmable approach, where the user can develop his own software for all the system components and organize his own data flows and image processing sequences.

3 Parallel Architectures for Computer Vision

Sophisticated vision tasks like those mentioned in the introduction imply numerous requirements for a vision system, which have been implemented into the systems to be described:

- Bottlenecks should be avoided either by extending or by specially designing the system. Addition of supplementary modules for parallelization and the support of general purpose processors with special hardware processors help to remove these bottlenecks.
- Image acquisition should be separated from the image processing and interpretation steps (see figures 1 and 2) in order to process data at maximum speed.
- Organization of data should not be fixed in order to give the system the greatest flexibility.
- With respect to signal acquisition, storage and display, different sensors must be available, differing in format but also in physical nature, in order to be able to process multidimensional images and multisensory signals.
- Processing of images, isolated, endless or in sequence, should be possible. Furthermore, the system hardware must be able to adapt to particular classes of tasks, varying in complexity, in order to allow a cost-effective match of the system to the application to be solved.

Bus-oriented multiprocessor architectures whose modules are the processing elements can efficiently be used. The modules can be classified into three categories:

1. Video I/O modules for acquisition, storage and display,
2. Pixel oriented image processing modules working synchronously with the image acquisition rate and offering the possibility of parallel and pipelined processing, for data compression

Figure 2: Processing levels in machine vision.
in the iconic stage (image processing and feature extraction),
not pixel oriented data processing modules working
asynchronously with the image acquisition rate, for statement
generation in the symbolic stage.

The system should offer the possibility of being used as a
development system (e.g., for testing the image processing or
evaluating the processing steps needed for solving a specific vision
task), or as part of a workstation (e.g., for the solution of real
problems in industrial environment). A hierarchical layered approach
can be retained for the software organization, which does not require
the user to be a low-level programming expert. Furthermore, an
exhaustive package of image processing and evaluation programs
should be provided. The development of problem-oriented software
should be possible using high level languages and ergonomic
man-machine interfaces helping the user in his application software
development.

The architecture of the parallel vision machines developed (see
fig. 3 for an example) implements the requirements stated in the
preceding section and is matched with the three processing levels
described in fig. 2. The image to be interpreted is sent to the low-level
processing unit. The result is then fed to the medium-level processing
unit which produces features for the high-level processing unit.

A feedback loop has to be foreseen from the high-level stage to the
medium and low-level stages (use of high-level knowledge to guide
the processing of the lower stages). The general architecture looks
like a system of pipelined processing elements, each of the processing
units being able to make use of its own internal parallelism, adapted
to the type of computation to be performed (leading to the concept of
multi-parallelism).

The Low-Level Processing Units. The low-level processing is
performed either with a VISTA system (Visual Interpretation System
for Technical Applications) developed at the Fraunhofer Institut
IITB in Karlsruhe, FRG [13, 16] illustrating the modular
functional design approach- or with dedicated hardware build around
a mesh of 1 bit-processors at ENSPS in Strasbourg. \( \Gamma \) [14, 15],
[16] exemplifying the reconfigurable, flexible and programmable
design approach.

The Medium-Level Processing Unit A network of sixteen T800
Transputers (Multicluster 2 from Parsytec or T800 Node from Telma)
has been chosen as a medium-level processing unit. The Transputer
technology has been retained because of its flexibility allowing to
efficiently accommodate the computation needs, as the complexity of
processing increases from the low level to the medium level. At this
level, OCCAM has been used as the programming language for
Transputers, because it provides the advantages of a high-level
language and permits to easily exploit concurrency. The OCCAM
compiler is also very well optimized as regards executable code size
and execution time. As the inputs of the medium-level stage are
images, an interface between the low-level stages video-bus and
Transputer links of the network has been developed to optimize data
transfer rates between the low and medium processing stages. The
integration of the Transputer based systems is shown on fig. 3. The
Integration of the specialized I/O interfaces, the low level processor
modules, the medium level Transputer networks and the cooperation
between the two levels is achieved through partitioning of the
processing algorithms over the two stages, taking into account the
characteristics of each processor and the kind of algorithms that each
processor can optimally execute. Consequently, the feasibility of an
efficient combination of the SIMD and MIMD approaches for low and
medium-level image processing has been confirmed. The processing
capabilities in actual applications has also been verified (see section
4.). This should lead to a tentative definition of the domains of use of
each level of parallelism, leading to rough rules for finding an optimal
architecture for a given application and/or a given time performance
requirement.

The High-Level Processing Unit. The high level processing unit is a
SCN 4.330 workstation. This workstation is receiving features from
the medium-level processing unit in order to interpret the initial real
world scene content. Programs running on this workstation are
written in C++, an object oriented programming language, which
allows a flexible and easy manipulation of the feature data coming
from the medium level stages and usually packed into specific
objects. The SCN workstation initializes, configures and loads the
two vision systems and the Transputer networks with appropriate
algorithms and is used as an interface between the machine vision
system and its environment (Ethernet, robot control...). Communication (and feedback loop) with the low/medium level
stages are done using either specially developed Transputer based
interfaces, or a Transputer board (BBKV2 from Parsytec) with dual-
ported RAM between the Transputer memory bus and the VME or
Multiplex bus of the low level stages, or a bus interface between the
workstation bus and the low level system busses. Communications
with the Transputer network are performed with the help of another
Transputer based board (VMTM from Parsytec).

When planning a vision system, it is necessary to provide universal
tools in order to relieve partly the user of the burden of programming
such complex systems. Due to the envisioned applications, a great variety of tasks results, leading to very distinct
operating facilities and system performance. Furthermore, in
laboratory, tools are needed for the development of methods and for
the implementation of algorithms in order to test their time and logical
behaviors. The tools must provide facilities enabling the user to take
advantage of the underlying hardware without special knowledge
about this hardware. On the other hand, the requirements for
industrial applications are high performance, short reaction time and
minimum user interaction. In our systems, image processing steps
can be run autonomously or interactively. The compiled code for the
processing steps is downloaded into the user program, where it is
parameterized and started. Once a step has been started, it continues
autonomously, independently of the calling user program and
synchronized with it via an autonomous control software. The
software structure is also in charge of the high-level unit. It can be
seen as the overall system management software, including the
man-machine interface. This man-machine interface is running on the
SUN workstation and is developed under X-windows and OSF/MOTIF,
which allow the development of menu-driven applications.

A Distributed (Transputer Network Based) Development Tool. Within the Parallel Computing Action Application PCA 4117, two
Transputer networks will be interconnected using an optical fiber.
The two systems, located the first in Strasbourg -France, and the
second in Karlsruhe -Germany (see fig. 3), will be interconnected
through an already existing optical fiber between the two institutes
involved in the application. The protocol chosen for data exchange is
the newly available FDDI standard. The performances of this
standard are such that the overall performance of the distributed
transputer network should not be lowered. The connection also
punctually allows the access by one institute to the whole shared
resources, when high computing power is required. The distributed
system can then be seen as a very complete development system for
vision applications, joining the resources of the two institutes and
enabling the use by each institute of the two different low-level
processing units.

4 Applications
The machine vision systems described are able to deal with quite a
lot of applications. A brief description of one of these follows. An
example of more sophisticated application, currently under
development and to be run on the described hardwares, is the
inspection using computer vision, on line in the manufacturing environment, of parts being manufactured [17, 18]. Inspection takes place through comparison of CCD images and corresponding computer generated images in the CAD model of the part. Comparison takes places as well at feature level as at image level. All kind of inspections, ranging from conformity checking up to metrology, can be achieved through use of an user friendly planning system. Due to the limited resolution of the imaging sensors used today for digital image processing, the sensor is moved in order to scan larger workpieces in their entire extent. CAD-based knowledge is also required to achieve efficient performance for each hybrid mechanical and electronic tasks. To inspect the geometrical properties, the image of the current field of view is compared with the information stored in the database of the associated CAD system, with or without use of structured light. Comparison takes place after segmentation and registration of the actual image with a synthetic projection of the CAD module. Furthermore, the 3D data coming from the CAD system are used to generate a 2D representation corresponding to the angle of view of the sensor. The output of the inspection stage is used for retrofitting by the manipulator in case of a possible remanufacturing.

The advanced inspection system described consists of knowledge processing and engineering at high level, but the first layers will be automated, dynamically optimized, data acquisition.

The manipulator in case of a possible remanufacturing. Figure 3 shows the diagram form in fig 3.

5 Conclusion

A model of machine vision architecture has been implemented using different functional levels, leading to a general purpose parallel computer for machine vision. The machines are based on pipelined low, medium and high level processing units, each having its own well-suited parallelism.

As a result of experimentation carried out with the vision described systems, the assessment of SIMD and MIMD parallel architectures for low level processing tasks and the development of knowledge based methodologies for high level applications, should allow improved performance and competitiveness in the development of wide range of image understanding applications, all based on two common, very flexible parts:

- A kernel of image processing routines for software. A general software kernel for vision systems would allow the designer to go from a specific application to the other only by changing the knowledge specific to the application (model of the scene under examination and its content), instead of changing the whole software.

- An heterogeneous parallel machine for hardware. The demonstrated feasibility of programmable hardware for image processing enables faster implementation of a wide range of industrial applications. With only one programmable parallel machine, it will be possible to solve a number of disparate computer vision applications related to various domains.

However, "other work is necessary in this field, as research must be continued with the aim in particular of building a better SIMD machine and of improving the coupling with the SIMD machines. In addition, it is necessary to develop an extensive software support, as software implementation of algorithms will definitively be the major cost factor in developing new applications. Furthermore, the future realization of vision systems will be influenced in particular by the integrability of vision systems as sub-systems in complex applications, possibly in the feedback path of the control loop (see figure 2) of these systems in order, in particular, to control the sensors and/or the lighting conditions (active sensing) for an automated, dynamically optimized, data acquisition.

Such vision systems are likely to run in a lot of applications, among which comparison of CCD and CAD data for automated workplace inspection was described. The type of computer vision architectures presented seems to be very promising for the future, as artificial vision is becoming a more and more important. The approach described could also lead to more "intelligent" robots, using such vision systems to understand their environment.

Acknowledgements

The authors are greatly indebted to Prof. Dr. H.-H. Nagel for having provided them the insight in computer-vision. They also wish to thank their colleagues at the ENSPS/LSIT and Fhg-ITB Institutes and for the kindful permission to use some of the illustrations presented in this paper.

References

PARALLEL ALGORITHMS FOR LOW LEVEL VISION ON A CONNECTION MACHINE (CM2)
Josiane Zerubia and Florimond Ployette

Abstract
In this paper, we show how data parallelism can be used for micro-control two low level vision algorithms based on deterministic relaxation techniques. First, we review the architecture and programming model of the CM2. Then, we give some details about the implementation. Finally, we compare both algorithms in terms of running time per iteration for different image sizes. We also present a comparison between parallel router and serial implementations of the same algorithm on a CM2 and on a SUN4 respectively.

1 The Connection Machine
In this section, we briefly describe the architecture of the Connection Machine, a more detailed description can be found in [3],[5]. The Connection Machine is a single instruction multiple data (SIMD) parallel computer with 8K to 64K processors. Each processor is a 1-bit serial processor, with 32K bytes of local memory and a 8MHz clock. The Connection Machine is accessed via a front end computer which sends macro-instructions to a microcontroller. All processors are commanded by the microcontroller in receiving the same macro-instruction at a given time from it. Physically, the architecture is organized as followed:

- The CM2 Chip contains 16 1 bit processors.
- A Section is the basic unit of replication. It is composed of 2 CM2 Chips, the local memory of the 32 processors and the Floating Point Unit.
- The interprocessor communication architecture is composed of two distinct networks:
  - A nearest-neighbour network, the NEWS Network (North-East-West-South), interconnects processors in groups of four.
  - A more complex network called the Router Network is used to provide general communication between any pair of processors. Each group of 16 processors is connected to the same router and each router is connected to 12 other routers forming a 12-dimensional hypercube.

For a given application, the user can dynamically define a particular geometry for the set of physical processors that has been attached.

The processor resource can be virtualized when the number of data elements to be processed is greater than the number of physical processors. In such a case, several data elements are processed on a single physical processor. Such a data-parallelism model architecture is well suited to computer vision as it is expressed in [4], [6].

2 Parallel implementation of relaxation algorithms
2.1 Mathematical model
We use a probabilistic model of an image based on Markov Random Fields (MRF). Two fields are used to model the image: one for the intensity to be restored, the other one for the discontinuities (edges). The problem is the approximation of a surface given noisy depth data on a regular 2D lattice of sites. The value of the intensity field at each site (i.e. each pixel) is given by the surface height at that site.

---

The authors would like to thank D. Ray and H. Voorhees from Thinking Machine Corporation for many useful discussions.
We suppose the noisy image is described by:

\[ d_{ij} = y_{ij} + n_{ij} \]  

where \( n_{ij} \) is a white Gaussian noise and \( y_{ij} \) is the original data.

The energy can be expressed by:

\[ E = D + S + P \]

where:

\[ D = \sum_{ij} (y_{ij} - d_{ij})^2 \]

\[ S = \sum_{ij} \lambda^2 ((y_{ij} - y_{i-1,j})^2(1-l_{ij}) \]

\[ + (y_{ij} - y_{i,j+1})^2(1-m_{ij})) \]

which models the smoothing constraint \((\lambda^2 * y + adiabatic)\) and:

\[ P = \sum_{ij} \alpha (l_{ij} + m_{ij}) \]

which is the cost to pay for introducing an edge.

The problem is reduced to the minimization of a non-convex energy function. Usually two kinds of techniques are used to solve this problem:

- stochastic techniques such as Simulated Annealing,
- deterministic techniques such as Graduated Non Convexity, Mean Field Annealing, Iterated Conditional Mode.

We are interested in deterministic algorithms and their parallel implementation.

2.2 The algorithms

In this section we describe two deterministic algorithms for edge detection and image restoration: the graduated non convexity technique (GNC) originally proposed by Blake & Zisserman [1] and the mean field annealing (MFA) used by Geiger & Giresu [2] and Zerubia & Chellappa [7].

For the GNC the basic idea is the following:

- The first step is to build a convex approximation \( E^* \) of the energy \( E \).
- The minimisation of \( E^* \) gives a global minimum.
- Then a sequence \( E^{(p)} \) is built so that \( E^1 = E^* \) and \( E^{(p)} = E \) for \( p = 1, 1/2, 1/4, 1/8 \).
- The minimisation of \( E^{(p)} \) uses as initial conditions the result of the minimisation of \( E^{(p-1)} \).

The MFA algorithm is based on Mean Field approximation used in statistical mechanics. This approximation consists of replacing the stochastic interaction among the fields at different locations by the interaction of the field at each site with the mean field values at different locations. Once the partition function \( Z \) has been approximated, it is easy to derive a set of deterministic equations for the mean field values \( \bar{y}, \bar{l} \) and \( \bar{m} \) of the intensity and the line-processes.

2.3 Parallel implementation

Both methods are based on the weak membrane model (cf.(2)) and both algorithms are inherently serial: each step produces a pixel map which is taken as input for the next step. For the GNC, we implement a checkerboard version of the successive over relaxation method [1] to minimize the energy and for the MFA, we use an optimal step conjugate gradient descent [7].

Although both techniques are deterministic, it takes a lot of computational time on a sequential computer to get the edge map and the restored image. Our attempt in reducing this time is based on the fact that, in early vision processing, much of the time is spent in performing local computation. Herein, we use data parallelism (one pixel per virtual processor) and fast local communications (NEWS) provided by the underlying architecture [3, 5]. For global operations like computing the energy value over the whole image, we use the reduce primitives.

3 Experimental results

These two algorithms have been run on a great variety of images. The discussion that follows is about their running time on the CM2 using 8K processors. Qualitative aspects are discussed in [8].

Table 1 shows the number of different types of instructions for each algorithm. All these instructions are 32 bit floating point instructions. The interesting point is that the interprocessor communications are local and the ratio between the number of communication instructions and the number of other instructions is very low.

<table>
<thead>
<tr>
<th>Arith.and Compar.</th>
<th>NEWS</th>
<th>Global Ops.</th>
</tr>
</thead>
<tbody>
<tr>
<td>GNC</td>
<td>117</td>
<td>10</td>
</tr>
<tr>
<td>MFA</td>
<td>123</td>
<td>24</td>
</tr>
</tbody>
</table>

Table 1: Number of CM instructions per iteration.

The GNC and MFA parallel algorithms implemented on the Connection Machine are compared. For each one, we give:

- The Virtual Processor Ratio (VPR) i.e. the number of virtual processors per physical processor.
- The CM time.
- The total time.
- The number of iterations for each algorithm.
- The CM time per iteration.

In table 2 and 3 are displayed the results for two images: a noisy aerial image (cf.fig.4) and an infra-red image.
processor. However, as the number of pixels per processor grows, the ratio between the execution time per processor and the number of pixels per processor is a bit lower. This is due to the fact that, the higher is the VPR, the higher is the percentage of local communications.

<table>
<thead>
<tr>
<th></th>
<th>Fortran-Sun</th>
<th>*Lisp-CM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total time</td>
<td>Time per it.</td>
<td>Total time</td>
</tr>
<tr>
<td>GNC</td>
<td>15mn.</td>
<td>1.66s</td>
</tr>
<tr>
<td>MFA</td>
<td>25mn.</td>
<td>7.97s</td>
</tr>
</tbody>
</table>

Table 4: Noisy Aerial Image 128x128

![Figure 3: Noisy aerial image 128x128, (SNR=5db)](image)

![Figure 4: Edge map with GNC](image)

![Figure 5: Noisy Aerial Image 128x128](image)

Figure 5: Execution time of the GNC and MFA Algorithms

4 REFERENCES


A FLEXIBLE MULTIPROCESSOR ARCHITECTURE
FOR HIGH-LEVEL COMPUTATION IN ELECTRONIC MEASUREMENTS

Alessandro GANDELLI  Vincenzo PIURI
Department of Electrical Engineering Department of Electronics
Politecnico di Milano piazza L. da Vinci 32, I20133 Milano, Italy

ABSTRACT - The massive computation requirements of an increasing number of applications in electronic measurements need dedicated hardware architectures to match the high throughputs of real-time environments. This paper presents some results about design and implementation of an integrated measurement system based upon a multiprocessor architecture.

I. INTRODUCTION

A number of high-computing applications are nowadays practically feasible and appealing due to the availability of realisable, reliable structures for massive computation at reasonable costs, e.g. in digital signal processing. In particular, it is possible to implement specialised architectures for management of large data in real time for extracting a reduced amount of essential information. Complex and accurate data observation and system control may be implemented by using this characteristic figures of the input data. A great number of examples may be found in electrical and electronic areas: one of the most important applications that becomes now realisable is constituted by the systems for real-time measurement for power and armonic control in power distribution.

In this paper we present a new integrated approach to real-time measurement of electrical parameters which is based upon a multiprocessor system. Such architecture allows to satisfy the massive throughputs and requirements of complex, high-level measurements. In particular, we present the overall hardware architecture and the dedicated software environment for developing application-specific measurement algorithms.

Our parallel architecture is composed by a SIMD multiprocessor unit system for parallel computation and by a front-end for an easy user interaction and control. Both such components are connected to a standard VME bus and to a dedicated control bus. The front-end computer is a standard Motorola's 68020 board, with mass-storage devices and input/output subsystems. Such computer provides the high-level human interface for designing, testing and executing the application-specific measurement algorithms. The front-end computer allows to load the measurement algorithm in the program memory of the multiprocessor subsystem, to overview and control the computation in such subsystem.

II. APPLICATION AND MATHEMATICAL REMARKS

Electronic measurements require execution of high-computing algorithms to extract basic characteristics and features from a large amount of data. Usually, when the measurements are used as information sources for choosing the suited actions in plant or process controls, all operations must be performed in real-time. This constraint makes impressive the amount of data that must be treated on-line during system management.

One of the most recent and important applications in the fields of electrical engineering is the real-time accurate control of high-voltage transmission and distribution systems. In this application, real-time electronic measurements are used to compensate the armonic contents of current and voltages due to the characteristics of the distribution system and to the electric loads. The basic aim of such operation is in fact the optimisation of power distribution with minimisation of power losses. Real-time control is important to measure dynamically the armonics and to provide continuously updated feedback information to the armonic compensator. This power electronic system suppresses unexpected and undesired frequencies and corrects reactive power.

High computational capabilities and possible parallelisation of the basic computation are two basic features to implement efficient architectures for real-time applications when massive computation is concerned [3], [4]. In the application case presented above, these characteristics allow to guarantee a very low response time in driving the electric power system for armonic compensation and, thus, to achieve an effective and optimal power distribution.

In this application, there is also a very large set of input data, which are composed by the samples of three-phase voltages and currents. Since control activities operate only on integral forms of such inputs, the high-level management of the system does not need a detailed and complete knowledge of the samples' values. In fact analysis of the electrical behavior of the distribution system can be performed by extracting the characteristic figures both in time or frequency domains. A pre-processing and compacting function must therefore be executed to reduce the quantity of information. This is useful to allow acquisition, manipulation and storing only of the necessary data in the control computer. Data compression and reduction are an important step towards the implementation of real-time control of complex systems at reasonable costs.

To satisfy strict computational requirements, the computer architecture must be able to guarantee a high computational power and, above all, an impressive input throughput. Output is generally very smaller than input flow, due to the characteristics of the computation input/output management is thus strongly unbalanced towards input activities. Neither traditional computers nor known architectures of parallel processors [3], [4] are able to provide a flexible and high-throughput connection between the processing units and the external environment, at least at reasonable costs. Therefore, a new dedicated solution, as the one we propose, is needed to achieve an efficient system without using expensive structures.

By looking at the algorithms used in electronic measurement and in system controls, we can identify a low number of typical mathematical operations that are common to the most of the algorithms. Traditional arithmetic operations are obviously widely used in all numerical algorithms. Therefore, the architecture, specialised for measurements, must greatly optimise execution of data transfer, addition, subtraction, multiplication and division.

A second group of more complex operations constitutes the basic kernel of a large number of algorithms to compute electrical quantities: the data transformations. Different kinds of data transformations may be identified in the most of the popular algorithms. Some examples are Fourier, Walsh-Hadamard, sine and cosine transformations, correlation and cross-correlation [5], [7]. These algorithms allow to extract the different spectral contents of the input data and to generate composed control signals. The previous mathematical operations require execution of many simple arithmetic operations (e.g. additions, subtractions and multiplications) [7], efficient generation of the corresponding coefficients and proper data addressing and use. An effective implementation should therefore be tailored to perform the most of such activities directly in hardware to increase the overall computational power of the architecture. Configurable devices should be preferred to hard-wired solutions to guarantee the flexibility of the system and adaptability to different specific applications and measurements.

III. HARDWARE ARCHITECTURE

The parallel architecture, presented in this paper, has been designed especially for advanced measurement applications. Its features are thus strictly connected with the specific algorithms implemented in measurement procedures. The processor design has been studied to
obtain the best performances not only for some traditional arithmetic operations, but also for high-speed digital signal processing operations (e.g., Walsh and Fourier transforms).

Our parallel architecture is composed by a multiprocessor subsystem and by a front-end computer, as it is shown in fig. 1. The first one allows to execute high-speed parallel computation, while the second one provides an easy user interaction and control. Highly-parallel computation is supported by using the 32-bit VME system bus and a 70-bit dedicated bus for instruction transfer and broadcasting. High throughputs are achieved by using dedicated 16-bit data busses (one for each processing unit) which connect the multiprocessor subsystem to the external data sources.

---

**Fig. 1 - The multiprocessor architecture**

Our structure differs from many other parallel structures since it behaves basically as a co-processing unit of the front-end computer. In fact it may be used to extract some characteristic figures from a large amount of input data by means of traditional signal processing algorithms. This pre-processing activity greatly reduces the quantity of data that must be treated by the front-end computer, in particular for applications in electronic measurements and automatic control.

The front-end computer is a standard Motorola’s 68020 board, with hard and floppy disks, printer, plotter, serial and parallel communication channels, graphic display. The main goal of this computer is the definition of a standard Unix environment for the user activities. Moreover, it provides the high-level human interface for designing, testing and executing the application-specific measurement algorithms. In particular, the front-end computer has been designed to support loading of the measurement algorithms in the multiprocessor bank and controlling of the parallel computation. The multiprocessor subsystem and the front-end computer are connected to a standard VME bus and to a dedicated control bus for data and control exchange.

The multiprocessor subsystem is a SIMD architecture, in which a single instruction is performed at the same time by all processing units [3][4]. The main goals of our design approach to a multiprocessor architecture were performance, modularity, expandability, flexibility and adaptability. Performance is an obvious requirement to implement structures for massive computation. Modularity allows to achieve a compact standard architecture in which the component can be modified and improved according to the requirements of the specific applications. Simple and fixed interfaces, interconnections and protocols guarantee the expandability of the structure (at least in a reasonable range) by addition of identical modules to match the algorithm characteristics. Flexibility and adaptability of the architecture to a large class of measurement algorithms is needed to implement a unique computing structure that may be used for a number of different applications. Adaptability should hold also at run-time by means of a software reconfiguration of interconnections to guarantee a high availability and flexibility of the system. Many of these features were achieved by adopting a simple, clear, high-speed, modular architecture based upon standard components.

The multiprocessor subsystem is composed by identical processing units and by a controller unit (see fig. 1). The processing units provide the parallel computational power by working on different data sets contemporaneously. The control unit has been introduced to overview the multiprocessor activity by imposing the sequence of instructions that must be performed by all processing units in parallel. The number of processing units depends on the specific measurement algorithm required by the user. We have found from extensive experiments that a good balance between the multiprocessor performance and host computer operativity in the minimal configuration can be achieved when the units are no more than 8.

Each processing unit is composed by two boards: the ALU (Arithmetic Logic Unit) board and the local memory. The ALU board is the heart of the computational structure to implement high-speed algorithms for electronic measurement. The memory board is basically used to store measurement data coming from the external environment, during execution of the application-specific algorithm. Performing data transfer inside each board is guaranteed by an internal high-speed bus for data and addresses. Our implementation of the experimental multiprocessor, discussed in this paper, considers arithmetic units for 16-bit data and operates at 10 MHz to avoid hardware complex structures and to provide high flexibility, powerful computational capabilities and high-throughput data management at reasonable costs. The structure of the processing unit is given in fig. 2.

---

**Fig. 2 - The processing unit**

The ALU board is a custom structure for execution of basic arithmetic operation and some complex arithmetic kernels of digital signal processing. The ALU board is based upon AMD's 29K01 ALU's and AMD's 29K17 multiplier.

Two three-port VLSI ALU's perform the computation on 16 bit data using a multi-bus and multiregister architecture. A high perfor-
mance 16x16 bit multiplier completes the arithmetic hardware and provide the suited computational support for a high-speed implementa-
tion of FFT algorithm. The two ALUs can perform multiple data operations during the same clock cycle using their multiport struc-
ture (two input-output ports and one input port) and the six register
operating independently by the internal ALU and MUX registers.

The memory board has been designed to obtain the best per-
formance and to reduce data transfer time without limiting system
capabilities. This board is strictly connected to ALU board. Never-
thess, it has high autonomous capabilities for data management. A
double memory bank has in fact been developed to allow contempo-
rary memory management from both internal processor resources and
system ALU. Every bank is formed by four 8x32 kbit memories con-
ected to the internal 32-bit bus. These two bus can be alternatively
switched to one of the three possible connection paths (ALU's, direct
I/O, system bus).

Two independent data paths are provided by the memory board
in order to exchange information with the external environment.
- the main data path, which is connected to the system bus,
- the secondary data path, that provides a direct input bus for data
acquisition by using the DMA technique.

While the processors are working on odd banks, the host computer
performs data transfer operations from/to the main memory for the
even banks, and vice versa. The switching system provides the control
of active paths in order to avoid bus conflicts.

When DMA techniques are adopted, at first the control logic exe-
cutes memory reading operations (performed by the main CPU) and,
then, DMA acquisition from secondary data port since data are writ-
ten in the same storage. Both these activities can be implemented
while the processor is working on the other bank because their execu-
tion time is smaller than time required to complete a Walsh or a
Fourier transform.

The sine and cosine generators (for FFT algorithm) or Walsh
function generator (for FWT algorithm) and latches providing the bus
switching during computational activities have been arranged on the
memory board to limit the ALU complexity.

As we have seen, the multi-access structure and management of
input data and memory banks are the most important novelty of our
approach to multiprocessor design for applications in electronic meas-
urements. The various possibilities of overlapping data input and
internal computations greatly increase the throughput and the compu-
tational power of the system to match the massive requirements
of real-time applications in measurement and control. While one half
of a processing unit is computing the nominal algorithm, the other is
supporting the input/output mechanism.

Parallel computation and management of the boards of the pro-
cessing units must be performed by activating the proper control sig-
als on the dedicated control bus. Due to the complexity of the dif-
f erent activities that can be executed in parallel and to the number of
control signals, direct control from the front-end computer is not
feasible.

To provide an efficient mechanism for operation fetching and de-
coding in the multiprocessor system, we defined a set of instructions
that can be executed by our architecture. Each instruction corre-
sponds to the compact coded representation of the control signals that
must be activated to impose execution of the desired operation. A
program for the multiprocessor system consists of a sequence of such
instructions.

Since our multiprocessor is a SIMD machine, only one control unit
is needed to supervise and manage all processing units. This board is
composed by the program memory and the processing controller. The
program memory contains the sequence of instructions (in the coded
format) that the multiprocessor must execute. The processing control-
ner performs the fetch and decode operations of the multiprocessor
program.

The processing controller is an AMD's 2910 microcontroller with
its own control circuits, private memory and latches. The microcon-
troller fetches and decodes the operations for the multiprocessor. For
each new instruction, it acquires the coded representation from the
program memory and identifies the current instruction. Then, it gen-

IV. SOFTWARE ENVIRONMENT

The use of an advanced hardware architecture generally presents
a number of practical difficulties, which reduce the usability of the sys-
tem and a complete exploitation of the new features. In our architec-
ture, this drawback is essentially related to the complexity of the struc-
ture and to the non-traditional parallel programming style. To avoid
this drawback and to give access to new computational paradigms also
to non-expert users, an integrated environment become necessary.

In our research we developed different tools which guarantee a
high level view to the hardware architecture and easy programming
of a large class of numerical-intensive applications. A general schema
of the software environment is shown in fig. 3. The main entities and
tools we developed and integrated in the standard Unix environment
of the front-end computer are:
- the low level multiprocessor assembler and the related translator,
- the assembler run time library for direct access to all features of
the system,
- the high-level language C and the related compiler,
- a high-level run-time library for traditional programming in C
language,
- a program loader,
- a symbolic debugger for C language,
- an interactive control environment.

![Fig. 3 - The software environment](image-url)
To allow the user programming of application-specific algorithms for the multiprocessor architecture, it is necessary to define the sequences of instructions in the assembler of the multiprocessor itself. Such instructions are the coded representation of the control signals which impose execution of the desired sequences of operations. Whenever the multiprocessor controller fetches a new instruction, it decodes the compact representation and generates the proper signals to activate the devices required by the operation.

The symbolic representation of the assembler instructions of our multiprocessor is similar to all traditional assembler languages of microprocessors [1]. An instruction is given by the n-tuple

\[ (opcode, op-1, op-2, \ldots, op-n) \]

where \( opcode \) is the operator (i.e., the symbolic identifier of the desired operation), while \( op-i \) are the needed operands (i.e., immediate data, registers, memory addresses). The complete definition of the assembler language is given in [2]. On the front-end computer, we developed a translator that generates the machine code from the assembler source code of the user program. Parallelism and data transfer are demanded to the proper design of the assembler program.

To write new application programs in the multiprocessor assembler in a simple way, we provide a modular, expandable run-time library. It contains the most used basic mathematical functions (e.g., trigonometric, exponential and hyperbolic functions) and the advanced DSP operations (e.g., DFT, FFT, IFFT, convolution, correlation). Such functions are coded as efficient as possible by exploiting all hardware parallelisms.

For loading, debugging and executing the machine code, the front-end provides a user-friendly environment. The user can select the program file that must be loaded, can transfer such program in the memory of the multiprocessor subsystem, can run the program. Possibly, he can also execute the multiprocessor program in a step-by-step mode or by introducing break-points to observe the behavior of the system and to correct errors in algorithms under development.

A higher programming interface may be achieved by adopting a standard programming language and custom libraries for multiprocessor operations. In our system we consider at the moment a subset of the C standard programming language. Control and programming of the multiprocessor architecture is allowed by a run-time library that we designed and implemented for the most common operations. The user writes his programs as on a traditional monoprocessor computer and to correct errors in algorithms under development.

A flexible multiprocessor architecture has been presented in this paper, particularly suited for on-line processing of a large amount of data signals. The most original characteristics of such architecture is the high capability of acquiring data from the external environment due to specialized high-speed data paths for concurrent I/O operations. A powerful support to direct signal processing is the availability of dedicated hardware inside each processing unit for high-level mathematical computations (e.g., DFT). The traditional FFT algorithm can be implemented in our architecture by using ten instructions they are iterately executed for a number of times equal to the number of butterflies required by the computation. If the number of sampled points is \( N = 2^k \), the total execution time \( T \) for the FFT in a single processor board is \( T = 107kN/2 = 57k^2 \), where \( T \) is the clock period of the system. In our experimental implementation, a 1024-points FFT is executed about in 5 ms.

As software aspects are concerned, our system provides an integrated environment for development of application programs. A library of basic mathematical operation has been implemented. It is continuously updated and expanded to cope with the requirements of new applications we are experimenting in the area of electronic measurements.

Further researches are presently in progress to improve the abstraction level for the programmer of applications. In particular, we are studying a user-friendly interface and programming style based upon icons: non-expert users will create the computational graph, which describes operations performed upon input data, through graphical interaction with the development environment. The user will select functional building blocks from a menu and connect them into the computational graph.

A second research topic is the automatic identification of parallelism in the user program. Compiler and assembler translator should be able to describe a sequence of simple operations affecting disjoint sets of devices in the same processing unit by means of a unique instruction. Whenever group of operations can be executed at the same time, the compiler and the assembler must detect this event and collapse the operations into the unique instruction.

\section*{REFERENCES}


\textsuperscript{769} This work was supported by the CNR grant No. 90.00332.PF67
ON ASYMPTOTIC RELIABILITY FUNCTIONS FOR LARGE SERIES-PARALLEL SYSTEMS

KRZYSZTOF KOŁOWROCKI
Department of Mathematics
Maritime University
Gdynia, Poland

Abstract - A paper presents some theorems giving sufficient conditions for a reliability function to be an asymptotic of large series-parallel and parallel-series systems.

I. INTRODUCTION

We are interested in a wide class of systems which every two elements are connected to each other either parallel or series. In the investigation of these systems it can be noticed that their elements can be arranged and numerated this way that the system lifetime \( X \) is given by

\[
X = \sum_{i=1}^{m} a_i \times \prod_{j=1}^{n} \left\{ \sum_{i_i_1 \ldots i_i_m} \left( X_{i_i_1 \ldots i_i_m} \right) \right\}
\]

or by

\[
X = \prod_{i=1}^{m} a_i \times \sum_{j=1}^{n} \left\{ \sum_{i_i_1 \ldots i_i_m} \left( X_{i_i_1 \ldots i_i_m} \right) \right\}
\]

where

\[
\mathcal{X} = \left\{ X_{i_1 \ldots i_m} : i_1 = 1, 2, \ldots, k, i_2 = 1, 2, \ldots, l_2, \ldots, i_m = 1, 2, \ldots, l_m \right\}
\]

is the arranged family of the random variables corresponding with the lifetimes of the particular elements. In a first case the system is called a parallel-series system of order \( m \) and in a second case it is called a series-parallel system of order \( m \).

For all \( i_1, i_2, \ldots, i_m \)

\[
\begin{align*}
(1)_{i_1 j_1} &= (i_1 j_1, i_2 j_2, \ldots) = (1)_{i_1 \ldots i_m j_m} = k
\end{align*}
\]

then parallel-series and series-parallel systems are called regular. Moreover, if random variables of the family \( \mathcal{X} \) have the same distribution function \( F(x) \), i.e. elements have the same reliability function \( R(x) = 1 - F(x) \), then these systems are called homogeneous.

Now, assuming \( k = k_n \) and \( l = l_n \), where \( n \) tends to infinity and \( k_n \) and \( l_n \) are natural numbers, we obtain sequences of parallel-series and series-parallel regular homogeneous systems of order \( m \) corresponding with the sequence \((k_n, l_n)\). For these sequences of systems there exist sequences of reliability functions \( R_n^{(m)}(x) \) for parallel-series and \( R_n^{(m)}(x) \) for series-parallel system.

II. ASYMMETRIC RELIABILITY FUNCTIONS FOR LARGE SERIES-PARALLEL SYSTEMS

Definition 1.

A reliability function \( R(x) \) is called an asymptotic reliability function of a sequence \( R_n^{(m)}(x) \) or an asymptotic of the regular homogeneous parallel-series system of order \( m \) if there exist constants \( a_n^{(m)} > 0 \), \( b_n^{(m)} \in (-\infty, \infty) \), such that

\[
\lim_{n \to \infty} n R_n^{(m)}(a_n^{(m)} x + b_n^{(m)}) = R(x)
\]

at all \( x \) where \( R(x) \) is continuous.

770
Similarly we define an asymptotic of the regular homogeneous series-parallel system of order $m$. 

Theorem 1. Let
\[ d = \sup \{ x : \IR(x) > 0 \}. \]

If $\IR(x)$ is continuous at point $d$ in case when $d < -\infty$ and sequences $(a_n^{(m)}, b_n^{(m)}), (k_n, l_n)$ have the following properties
\[
\lim_{n \to \infty} \frac{1}{n^{m-1}} k_n = 0 \quad \text{for } m \geq 1
\]
\[
\lim_{n \to \infty} \frac{1}{n^{m-1}} k_n + \cdots + 1 - \IR(x) \leq \IR(x)
\]
\[
\leq \ln \IR(x) \quad \text{for } m \geq 1
\]
at all $x \in (-\infty, d)$ where $\IR(x)$ is continuous, then $\IR(x)$ is an asymptotic of the regular homogeneous parallel-series system of order $m$ for all $m \geq 1$.

Proof. ([4]).

Theorem 2. Let
\[
d = \inf \{ x : \IR(x) < 1 \}. \]

If $\IR(x)$ is continuous at point $d$ in case when $d > -\infty$ and sequences $(a_n^{(m)}, b_n^{(m)}), (k_n, l_n)$ have the following properties
\[
\lim_{n \to \infty} \frac{1}{n^{m-1}} k_n = 0 \quad \text{for } m \geq 1
\]
\[
\lim_{n \to \infty} \frac{1}{n^{m-1}} k_n + \cdots + 1 \leq \IR(x)
\]
\[
\leq - \ln \IR(x) \quad \text{for } m \geq 1
\]
at all $x \in (d, \infty)$ where $\IR(x)$ is continuous, then $\IR(x)$ is an asymptotic of the regular homogeneous series-parallel system of order $m$ for all $m \geq 1$.

Proof. ([4]).

IV. REFERENCES


III. CONCLUSION
AUTOMATIC 3-D HIERARCHICAL SUBSTRUCTURING SCHEME AND ITS IMPLEMENTATION ON A VECTOR-CONCURRENT MACHINE

MUKUL SAXENA
Corporate Research and Development
General Electric Company, Schenectady, N.Y. 12301

AND

RENATO PERUCCHIO
Department of Mechanical Engineering
University of Rochester, Rochester, N.Y. 14627

Abstract This paper presents an extension to the use of generic algorithms for Recursive Spatial-Decompositions (RSD) to design a hierarchical substructuring (HIS) scheme that can be easily coupled to the automatic mesh generator and can be embedded in a self-adaptive meshing-analysis procedure. The geometric algorithms for RSD generate nested-dissections (analytical substructures) of the domain that are used for the analysis scheme (HIS) described in this paper. An evaluation of the HIS scheme for parallel processing is discussed with reference to the results of implementation of the scheme on a vector-concurrent machine.

1 INTRODUCTION

This paper describes a 3-D FEM analysis scheme based on Recursive Spatial Decompositions (RSD). The scheme forms an integral part of an automated FEM meshing-analysis system that is ideally suited for parallel computing [1]. Specifically, spatial decompositions are used to (1) transform automatically a solid model into a finite element mesh, (2) perform incremental analysis via hierarchical substructuring, (3) produce hierarchical data structure that allows coupling between meshing and analysis, and (4) perform incremental self-adaptive analysis. The parallelism in the meshing scheme has been earlier described in [1]. In this paper the applicability of the hierarchical substructuring scheme is discussed in the context of parallel processing environments. A brief overview of RSD based meshing algorithm that forms the basis of the substructuring scheme is given in the following section.

2 AUTOMATIC MESHING SCHEME

A two-stage automatic meshing procedure is used for automatic generation of the finite element meshes. For a detailed description see Reference [2]. In Stage 1, the solid to be meshed, \( \Omega \), is approximated by a collection of variably sized octants through the recursive spatial decomposition of the original geometric domain. Such an approximation is conveniently represented by a logical tree structure whose nodes have eight sons, popularly known as “octree”. Stage 2 of the meshing algorithm transforms the RSD into a valid FEM mesh, through further processing of the individual octants.

Each node of the octree represents an informationally complete subdomain \( \omega_i \) and, in terms of the finite element model, a substructure. Thus, for the purpose of analysis, the octree can be regarded as a cataloging structure with geometrical and analytical information directly mapped on to it – see Figure 1 for a 2-D example. Such a substructuring scheme, whereby all the substructures are hierarchically organized and derived through the recursive spatial decomposition (RSD) of the original domain, is referred to as hierarchical substructuring. Note that the analytical substructures produced by the RSD are equivalent to the nested dissections described in [3].

3 HIERARCHICAL SUBSTRUCTURING

There are three distinct stages in hierarchical substructuring.

In the first stage, stiffness matrices are formulated for the lowest level nodes of the octree which consist of assemblies of linear isoparametric elements. The stiffness matrix formulation is based on the degeneration of the eight-noded isoparametric brick element.

Second stage is the assembly stage of analysis. Starting from the bottom of the tree, the stiffness matrices of the offspring of the same parent cell are assembled into a substructure and the interior degrees of freedom are eliminated by static condensation. Once all the substructures at a given level are assembled and condensed, the procedure begins to operate on the tree level immediately above.

Each level of the octree represents a substructure which in turn is composed of substructures from the lower level. This process continues until the root node of the tree is reached. The mathematical formulation of this stage can be described as follows: Let \( b \) and \( i \) denote respectively the nodes on the boundary and the interior of the substructure. The equilibrium equations for the substructure can be partitioned as

\[
\begin{bmatrix}
K_{ii} & K_{ib} \\
K_{bi} & K_{bb}
\end{bmatrix}
\begin{bmatrix}
X_i \\
X_b
\end{bmatrix} =
\begin{bmatrix}
R_i \\
R_b
\end{bmatrix},
\]

where \( X \) is the displacement vector, \( R \) is the load vector, \( K_{ii} \) are the stiffness coefficients related to the interior nodes alone, \( K_{bb} \) the coefficients for the nodes on the boundary and \( K_{ib} \) the cross-coupling terms linking boundary nodes to the nodes in the interior. By eliminating the interior degrees of freedom a new system of linear equations is obtained:

\[
K_{bb}X_b = R_b
\]

where \( K_{bb} = K_{bb} - K_{ib}K_{ii}^{-1}K_{ib} \) and \( R_b = R_b - K_{ib}K_{ii}^{-1}R_i \) are the reduced stiffness matrix and the reduced load vectors, respectively.

The third stage involves root level solution and recovery of displacements and stresses. The reduced system of equations, obtained at the end of assembly stage, is solved for the unknown boundary displacements, \( K_{bb} \). Once the root level displacements \( X_b \) are known the interior nodes are computed by solving the system

\[
K_{ii}X_i = R_i - K_{ib}X_b.
\]

Recall that \( K_{ii} \) has already been reduced to an upper triangular matrix.
during assembly stage. Thus $X_1$ can be derived directly through backward substitution in Equation 3. Finally, the displacements at the element level are used to compute the stresses in the solid.

$$\sigma = DB\delta^e,$$

where $D$ is the material property matrix, $B$ is the strain-displacement matrix and $\delta^e$ is the elemental displacement vector.

4 PARALLEL IMPLEMENTATION

Four domains were meshed using the octree-based mesh generator described in Section 2, and analyzed via hierarchical substructuring. Figure 2 shows the meshes used, the applied constraints and loads, as well as the deformed shape.

![Meshes and deformed shapes](image)

**Figure 2: Analysis via hierarchical substructuring - original meshes and deformed shapes for (a) block, (b) housing, (c) cyl_cyl_int, and (d) bracket.**

To study parallelism, it is convenient to identify three stages of varying computational complexity as follows: 1) substructures assembly and condensation except the root level (this includes computation of element stiffness and condensation of all the substructures up to level 1 in the octree); 2) solution of the final root level substructure; and 3) recovery of interior d.o.f.'s. Stages 1) and 2) are the dominant factors in terms of solution time and, therefore, only these two stages are considered for parallel processing. Stage 2 of processing consists of assembling the eight level-1 substructures and then solving the final root-level substructure. As such, stage 2 must be performed on a single processor. However, since it requires the reduction of a large, nearly fully populated matrix, stage 2 can take full advantage of vectorization.

In order to minimize the computational overheads associated with parallel processing (processor idle time) the following strategy was used for mapping the computational load to various processors:

1. Start at the lowest level of the octree and visit all the substructures at this level.
2. Identify the first non-empty substructure $S_j^*$ that can be assembled, i.e., all its sons are already assembled. (Hereafter, an asterisk denotes a substructure whose eight sons are already assembled.) Assign $S_j^*$ to the first available processor $P_1$. The next non-empty substructure $S_j^*$ is assigned to the second available processor $P_2$ and so on. Once all the eight sibling of an octree node are computed, the substructure corresponding to the parent node is marked as available for assembly.
3. The process continues until all the first level substructures are computed.

Table 1 shows the speed-ups and efficiencies for the example problems. $\rho$, $\eta$, $T_S$ and $T_P$ are the speed-up factor, efficiency, serial, and parallel time, respectively. All the times are reported for implementation on Alliant FX/8 machine configured for 8 processors. $\Delta\rho$ is the percent increase in the efficiency achieved by using the balanced processor load scheme, described above, in contrast to a scheme whereby eight level-1 substructures are assigned to separate processors. Notice that the increase in efficiency is marginal for the bracket problem. This can be ascribed to the fact that for this particular problem, condensation of one substructure dominates the whole execution cycle. Therefore, the processing time is only marginally influenced by the way other substructures are assigned to different processors.

<table>
<thead>
<tr>
<th>Object</th>
<th>$T_S$ (sec)</th>
<th>$T_P$ (sec)</th>
<th>$\rho$</th>
<th>$\eta$ (%)</th>
<th>$\Delta\rho$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block</td>
<td>9.19</td>
<td>1.229</td>
<td>7.52</td>
<td>94.0</td>
<td>41.0</td>
</tr>
<tr>
<td>Housing</td>
<td>23.30</td>
<td>6.154</td>
<td>3.79</td>
<td>74.0</td>
<td>45.0</td>
</tr>
<tr>
<td>Cyl_cyl_int</td>
<td>12.74</td>
<td>1.727</td>
<td>7.38</td>
<td>92.3</td>
<td>85.0</td>
</tr>
<tr>
<td>Bracket</td>
<td>85.67</td>
<td>30.853</td>
<td>2.78</td>
<td>34.8</td>
<td>30.0</td>
</tr>
</tbody>
</table>

**Table 1: Speed-up and efficiency for balanced load scheme**

Table 2 reports the speed-ups produced by exploiting vectorization for solving the root level substructure. Notice that the efficiency of vectorization increases with the length of the associated vectors (i.e. the size of the stiffness matrix), as illustrated by the speed-ups for housing and bracket problem.

<table>
<thead>
<tr>
<th>Problem</th>
<th>$T_{total}$ (sec)</th>
<th>$T_{vector}$ (sec)</th>
<th>$\rho$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block</td>
<td>12.123</td>
<td>6.613</td>
<td>2.16</td>
</tr>
<tr>
<td>Housing</td>
<td>56.306</td>
<td>13.475</td>
<td>4.22</td>
</tr>
<tr>
<td>Cyl_cyl_int</td>
<td>3.360</td>
<td>1.650</td>
<td>2.15</td>
</tr>
<tr>
<td>Bracket</td>
<td>270.064</td>
<td>40.695</td>
<td>6.70</td>
</tr>
</tbody>
</table>

**Table 2: Speed-ups due to vectorization of root level solution on the ALLIANT FX/8**

Table 2 reports the speed-ups produced by exploiting vectorization for solving the root level substructure. Notice that the efficiency of vectorization increases with the length of the associated vectors (i.e., the size of the stiffness matrix), as illustrated by the speed-ups for housing and bracket problem.

5 SUMMARY

This paper has presented a view that emphasizes on the use of RSD for 3-D Hierarchical Substructuring (HIS) scheme that can be incorporated in a parallel FEM analysis system consisting of automatic meshing, analysis via HIS, and self-adaptive incremental re-meshing and re-analysis. The octree structure provides a tight coupling between geometrical and analytical data, thus allowing for an efficient integrated meshing-analysis procedure [1]. In conclusion, the recursive formulation of the algorithms makes the scheme ideally suited for parallel processing.

References
A Parallel Algorithm for Clique Finding Problem*

Chain-Wu Lee
Institute of Computer Science and Information Engineering
National Chiao-Tung University
Hsinchu, Taiwan, R.O.C.

Shian-Shyong Tseng
Institute of Computer and Information Science
National Chiao-Tung University
Hsinchu, Taiwan, R.O.C.

ABSTRACT

In this paper, we try to solve clique finding problem, one of the most famous NP-Complete problems, in parallel. We propose a new sequential clique finding algorithm, the empirical and theoretical performances of the algorithm are further analyzed. The empirical results show that the average time complexity is about $e^{-2t}$, which grows in an exponential order. However, this algorithm can be easily parallelized if an MIMD shared-memory machine is used. Then a parallel clique finding algorithm is obtained. Therefore, the time complexity of our parallel algorithm is about $k \cdot e^{-2t}$.

1 Clique Finding Problem and Algorithm

The clique finding problem is defined as follows.

Definition. Given a graph $G = (V, E)$, the clique finding problem is to find every subset $V' \subseteq V$ such that every two vertices in $V'$ are joined by an edge in $E$.

Clique finding problem is one of the most famous NP-Complete problems. Generally speaking, exponential time is needed for it; however, there are some special cases which can be solved in polynomial [2] [4] [8] [9].

Next, we define some notations and terminologies used in this paper.

Degree of a vertex $v$ is $d(v); A(v)$ is the adjacent vertices of vertex $v$ whereas $A^-(v)$ is the adjacent vertices of vertex $v$. Clique $K(G)$ is a set of vertices which are mutually connected in the graph $G$, $K(G)$ is a set of cliques in the graph $G$ and all of the cliques in $G$ constitute of a set called $K_u(G)$. $K_u(G)$ is a clique which $v \in K_u(G)$; likewise, $K_u(G)$ is the set of cliques that all cliques in this set-include vertex $v$. The output of our algorithm is composed of two parts, essential part and optional part. Every vertex in the essential part is a clique and each combination of the optional part forms a clique with essential part.

If we give every vertex $v$ in graph $G$ a label, then $\lambda(v)$ is the integer value label of the vertex $v$. $G(v, \lambda^*)$ is said to be a restricted induced subgraph with respect to vertex $v$ in graph $G$. $G \sqcap v$ is an induced subgraph of $G$ and $(1) V' = \{v \mid A(v) = \{v' \mid v' \in \lambda \in \lambda(v)\}$, $(2) E' = \{(u, w) \mid u \in V', w \in \lambda(v) \in E\}$.

Now the notion of our algorithm is stated as below.

Given a graph $G$, let $K(G) = \{v_1, v_2, \ldots, v_m\}$ be a clique of $G$, $1 \leq m \leq n$ where $n$ is the number of vertices in $G$. It is trivial that any clique $k$ of $G$, for any arbitrary clique $k$, either $v \in k$ or $v \notin k$. The idea can be extended to a set of cliques, that is, for a vertex $v$, we have $K_v = K \cup K_k$, where $K_v = \{k' \mid k' \in k\}$ and $K_k = \{k \mid k \notin k\}$.

Second, if a vertex $v$ is connected to all vertices in graph $G$, then the cliques $K_v(G)$ may be either $\{v\}$ or $\{k \mid k = K_v \cup \{v\}, k \in K'(G')\}$, where $G'$ is derived from $G$ by removing $v$ and its adjacent edges. So we have the following lemma.

Lemma 1. If $v$ is a vertex which connects to all vertices in graph $G(V, E)$, then $K_v(G) = \{v\}$. $K_v = \{v \mid v \in A^-(v) \wedge v \notin \lambda(v)\}$ where $G' = (V - v, E - A(v))$.

From Lemma 1, if $v$ connects to all vertices in $G$, then $K_v(G)$ must be either $\{v\}$ or $\{v \cup K(G-V') \land V' - A^-(v)\}$.

If the vertices of the graph $G$ are labeled with positive integers, then we have the following lemma.

Lemma 2. $v \in K_v, u \neq v \wedge \lambda(v) < \lambda(u)$ then $k \subseteq V'$ where $V'$ is the vertices of $G'$.

Proof: Suppose the lemma is false. That is, $k \subseteq V'$, then there must exist at least one vertex $w \in K_v, u \wedge w \notin V'$, such that $\lambda(w) > \lambda(k)$ or $\lambda(u) < \lambda(k)$. For the former, it contradicts the definition of $G'$ because that if a vertex $w$ is not in $\lambda(v)$ or $w \in \lambda(u)$, then $w$ must be a vertex of $K_v$. For the latter, it contradicts the prerequisite of the lemma since $\lambda(v)$ is the smallest number among all vertices in $K_v$. Thus we can not exist in $K_v$.

According to Lemma 1 and 2, we have Theorem 1.

Theorem 1. $K(G) = \cup K_v(G_v)$.

Proof: Assume $K(G) = \cup K_v(G_v)$, and assume for $k \subseteq V'$, $k = \{u, v_1, \ldots, v_n\}$ and $k$ is with the smallest label among $\lambda(u)$. According to Lemma 2, we have $k \subseteq V'$, contradicting to our assumption.

This theorem, in other words, means that the cliques in graph $G$ can be found by discovering all the cliques in the restricted induced subgraphs of every vertex in graph $G$.

Next, we define some notations and terminologies used in this paper. memory pool, and exploit the algorithm recursively to find the solutions.

Second, if a vertex $v$, the processor $PE_i$ first picks up a vertex, say $v$, compute $G_v$. $G_v = \{v \mid \lambda(v) < \lambda(w) \wedge (u, v) \notin E\}$, push $G_v = \{v \mid \lambda(v) < \lambda(w) \wedge (u, v) \notin E\}$ into memory pool, and exploit the algorithm recursively to find the solutions.

If a processor $PE_i$, finishes the job, the subgraph pushed into the memory pool previously can be dispatched to the free processor $PE_i$ and repeats entire process.

The output of our algorithm contains two parts-an essential and an optional part. For the essential part, it is easy and straightforward, just output it. However, for the optional part, if we need to know every clique in the graph, we may adopt any arbitrary parallel combination generating algorithm on it and combine the output of the combination generating algorithm together with essential part.[l]

In the previous paragraph, we have not pointed out the labeling methodology used in our algorithm, but sometimes the labeling scheme may dramatically affect the load of the processor and the performance of the algorithm.

Thus we proposed the following labeling scheme. Label every vertex $v$ according to its degree $d(v)$ in ascending order, the smaller the degree of a vertex is, the smaller it labels.

By applying our algorithm and labeling scheme, we prune the smaller -cliques of the graph and left the largest clique not processed. This concept is based on the "Largest clique is the most time consuming part in the graph in computation".

2 Time Complexity Analysis of the Clique Finding Algorithm

In the following section, we try to analyze the time complexity of the parallel clique finding algorithm. Note that under the MIMD model when any processor is free, we assume it gets a job and no communication exists between any two subtasks. So we may suppose the speedup of the parallel algorithm is $n$ relative to its sequential version. Therefore, if the time complexity of sequential algorithm be denoted as $E_0(n)$ then the time complexity of parallel algorithm is certainly $E(n)/n$ for an $n$ processors machine.

The judgment criteria of the algorithm here we define is the number of cliques output. Therefore, according to our algorithm for random labeling scheme, first step a vertex $X$ is picked out for computation of the restricted induced subgraph. Suppose the degree of the picked vertex $X$ is $z$. Then the probability of the restricted induced subgraph $G' \subseteq \{n - 1\} \cdot p^2(1 - p)^{n-1}$.

For the restricted induced subgraph $G'$, a clique is output and then the vertex $X$ is deleted from it. The rest restricted induced subgraph will be recursively explored by the algorithm, so the time steps required for this algorithm is $E_0(p)$ for the rest restricted induced subgraph; therefore the time complexity for $G'$ part is $\{n - 1\} \cdot p^2(1 - p)^{n-1}E_0(p) + 1$ where $1$ is the time used for the essential part.

*This research was partially sponsored by the National Science Council under NSC70-0108-E009-017.
3 Simulation Results

Since the performance analysis of our algorithm with sorted degree labeling strategy is quite hard, therefore a set of simulation is held in the last section.

In the simulation, first, for every probability \( p \) and number of vertices \( n \), the experiment iterates 129 times. The range of \( n \) is from 1 to 25 step 2 whereas \( p \) ranges from 0.00 to 1.00. Graph representation for the simulation results is shown in Fig. 2. Semi-log graph for Fig. 2 is listed in Fig. 3.

After carefully examining the figure, the curve seems to be a straight line when \( n \geq 2 \), so the technique of linear regression is used to approximate this curve. Approximate line is also shown in Fig. 3. Undoubtedly, fitted curve is very proximate to the original curve. So the average time used in our algorithm is about \( E(n) = 0.148 + n + c \) where \( E_n \) is average time used for our algorithm, \( n \) is the number of vertices, \( c \) is a constant and 0.148 is the slope of the straight line.

Expanding the formula, we get \( E_n \approx k \cdot e^{0.27n} \).

4 Discussions

Maximum Clique Problem is to find a clique \( k \) where \( |A| \geq |k|, k \in K \). This problem is an NP-Complete problem.

A slight modification on our algorithm will be more efficient for the maximum clique problem. For sequential algorithm, first, we generate all restricted induced subgraph for the graph. Say, \( G_{x1}(3), G_{x2}(4), \ldots, G_{xn}(r) \) if there are \( n \) vertices in the graph; since maximum clique is the clique with greatest number of vertices in the graph, next step computation starts with \( G_{x1}(3) \) where the number of vertices of \( G_{x1}(3) \) is the largest among \( G_{x1}(3), G_{x2}(4), \ldots, G_{xn}(r) \), the above procedure will be repeatedly executed until the maximum clique is found. During the procedure we shall keep track of the size of vertices in the restricted induced subgraph generated. Once found the size of restricted induced subgraph generated is smaller than any other restricted induced subgraph \( G_{x1}(3) \), then the procedure jumps to process \( G_{x1}(3) \). This algorithm is somewhat similar to the breadth first search.

For parallel algorithm, we should keep a location for maintaining the current maximum clique size \( |k| \), and every processor gets a restricted induced subgraph with the same number \( |k| \) of vertices and then repeat the sequential algorithm.

References


NEW ALGORITHMS FOR POLYNOMIAL AND TRIGONOMETRIC INTERPOLATION ON PARALLEL COMPUTERS
ILAN BAR-ON AND AVRAM SIDI
DEPARTMENT OF COMPUTER SCIENCE, TECHNION, TECNION CITY, HAIFA 32 000, ISRAEL.

Abstract.
An interpolation polynomial of order \( n \) is constructed from \( p \) independent subpolynomials of order \( n \sim N/p \). Each such subpolynomial is found independently and in parallel. Moreover, evaluation of the polynomial at any given point is done independently and in parallel, except for a final step of summation of \( p \) elements. Hence, the algorithm has almost no communication overhead and can be implemented easily on any parallel computer. We give examples of finite-difference interpolation, trigonometric interpolation, Chebyshev interpolation, and conclude with the general Hermite interpolation problem.

1. Introduction. In this paper we study the problem of polynomial and trigonometric interpolation on large parallel MIMD computers. There are well-known sequential methods for both problems. However, these methods are not easily adaptable to parallel systems, and especially to loosely connected systems such as rings, stars, because of the overhead due to interprocessor communication, see for example[2, 3, 5]

In this work we present new algorithms for polynomial and trigonometric interpolation that require almost no communication between the processors. Given an interpolation problem of order \( N = np \), \( p \) being the number of processors, we divide it into \( p \) smaller interpolation problems of order \( n \). These problems are solved independently and in parallel using a appropriate sequential method. The value of the interpolation polynomial is then a combination of the corresponding subvalues.

2. The interpolation polynomial. Let \( f(z) \) be a function defined on \([a, b]\), whose values on the set of \( N + 1 \) distinct points \( X = \{x_0, x_1, \ldots, x_N\} \), is given by \( f_j \equiv f(x_j), j = 0,1, \ldots, N \). We are interested in constructing a representation of the polynomial \( P(z) \) of degree at most \( N \) that interpolates \( f(z) \) on \( X \) and is most suitable for parallel computation. Let \( \{X_1, X_2, \ldots, X_p\} \) be a partition of \( X \),

\[ X = \bigcup_i X_i, \quad X_i \cap X_j = \emptyset, \quad i \neq j. \]

The following theorem indicates how \( P(z) \) can be constructed independently and in parallel by \( p \) processors, each solving a smaller interpolation problem on one of the subsets \( X_i \).

**Theorem 2.1.** For \( i = 1, \ldots, p \), define

\[ w_{i,j} = \prod_{k \in X_i} (x_j - x_k), \quad x_j \in X_i, \]

and let \( Q_i(x) \) be the polynomial of degree at most \( \{X_i\} - 1 \) that satisfies the following interpolation conditions:

\[ Q_i(x_j) = w_{i,j} f_j, \quad x_j \in X_i. \]  

Then \( P(z) \), the interpolation polynomial on \( X \), is given by

\[ P(z) = \sum_{i=1}^{p} Q_i(z) \prod_{k \in X_i} (z - x_k). \]  

**Proof.** See[1]. \( \Box \)

It is known[4, 6, 7] that the barycentric representation for Lagrange interpolation enjoys a large scope of numerical stabili-

3. Finite Difference Interpolation. Let \( X \) be a set of equally spaced points in the interval \([a, b]\),

\[ x_i = a + ih, \quad h = \frac{b-a}{N-1}, \quad \text{for } i = 0,1, \ldots, N-1. \]

We assume for simplicity that \( N = np \) and \( p \) is the number of processors available. We consider here two partitions.

In the first partition we assign the \( i \)-th group of \( n \) consecutive points to the \( i \)-th processor,

\[ X_i = \{x_{ij} = x_{nj} + j, \quad j = 0, \ldots, n-1\}, \]

for \( i = 0, \ldots, p-1 \). Hence,

\[ w_{i,j}^{-1} = \prod_{k=0}^{n-1} (x_{nj} + j - x_k) \prod_{k=(j+1)n}^{n-1} (x_{nj} + j - x_k) \]

\[ = C_i(-1)^n \binom{n-1}{j} \binom{N-1}{in + j}, \]

776
where \( C_t = (-1)^{N-n}(N-1)!/(n-1)! \) is independent of \( i \) and \( j \). We note that if the same constant \( C \) multiplies all the \( w_{ij} \), it follows from (1), (3) that the subpolynomials \( Q_i(z) \) and \( R_i(z) \) are also multiplied by the same constant \( C \) but the interpolation polynomial \( P(x) \) remains invariant by (4). In view of this, each processor has to compute the corresponding polynomials \( Q_i(z) \) and \( R_i(z) \) that satisfy the interpolation conditions

\[
Q_i(z) = (-1)^{n-j} \frac{1}{j!} \frac{1}{(N-1)!} \sum_{k=0}^{n-1} j^{n-k} f_{i+j},  \quad R_i(z) = (-1)^{n-j} \frac{1}{j!} \frac{1}{(N-1)!} \sum_{k=0}^{n-1} j^{n-k} f_{i+j}.
\]

for \( j = 0, \ldots, n-1 \). As before, this computation can be carried out using any finite difference formula.

In the second partition the subsets \( X_i \) are formed according to

\[
X_i = \{ z_{ij} = z_{i+j}, \quad j = 0, \ldots, n-1 \},
\]

for \( i = 0, 1, \ldots, p-1 \). Consequently,

\[
w_{ij}^{-1} = \sum_{k=0}^{n-1} (z_{ij} - z_k),  \quad k = 0, 1, \ldots, N-1,
\]

where \( C_t = C_i / p^{n-1} \) is independent of \( i \) and \( j \). Each processor has to compute the corresponding polynomials \( Q_i(z) \) and \( R_i(z) \) that satisfy the interpolation conditions

\[
Q_i(z) = (-1)^{n-j} \frac{1}{j!} \frac{1}{(N-1)!} \sum_{k=0}^{n-1} j^{n-k} f_{i+j},  \quad R_i(z) = (-1)^{n-j} \frac{1}{j!} \frac{1}{(N-1)!} \sum_{k=0}^{n-1} j^{n-k} f_{i+j}.
\]

for \( j = 0, \ldots, n-1 \). As before, this computation can be carried out using any finite difference formula.

We have the following operation count for constructing and evaluating the polynomial, when \( p < N \):

<table>
<thead>
<tr>
<th>Operation</th>
<th>Sequential</th>
<th>Parallel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Construction</td>
<td>((N-1)N/2)</td>
<td>((n-1)n)</td>
</tr>
<tr>
<td>Evaluation</td>
<td>(2(n+p))</td>
<td>((n+p))</td>
</tr>
</tbody>
</table>

We obtain a speed up of order \( p^2/2 \) in the construction stage, and a speed-up of order \( p/2 \) in the evaluation stage as compared to the ordinary sequential finite difference methods.

4. Trigonometric interpolation. Let \( \theta_i \), be equally spaced points in \([0, 2\pi] \) given by

\[
\theta_i = \frac{2\pi i}{N}, \quad i = 0, 1, \ldots, N-1,
\]

and let \( f(\theta) \) be a function defined on \([0, 2\pi] \) whose values \( f_j \equiv f(\theta_j), \quad j = 0, 1, \ldots, N-1 \), are given. Furthermore, let \( N = 2M \). Then there exists a unique balanced trigonometric polynomial \( T(\theta) \) of degree \( M \),

\[
T(\theta) = a_0 + \sum_{k=1}^{M-1} (a_k \cos k\theta + b_k \sin k\theta),
\]

interpolating \( f(\theta) \) at the points \( \theta_j, \quad j = 0, 1, \ldots, N-1, \) see [1]. A complex interpretation of \( T(\theta) \) in terms of the variable \( z = e^{i\theta} \) yields the balanced complex trigonometric polynomial \( P(z) \equiv T(\theta) \) of degree \( M \),

\[
P(z) = \frac{1}{2} a_M z^{-M} + \sum_{k=0}^{M-1} c_k z^k + \frac{1}{2} c_M z^M,
\]

with \( a_M = c_M \), where their coefficients \( c_k \) are related to the \( a_k \) and \( b_k \), in (5) through

\[
a_k = c_k + c_{-k}, \quad b_k = i(c_k - c_{-k}), \quad k = 0, 1, \ldots, M.
\]

Of course, \( P(z) \) satisfies the interpolation conditions

\[
P(z_j) = T(\theta_j) = f_j, \quad z_j = e^{i\theta_j}, \quad j = 0, 1, \ldots, n-1,
\]

and this computation can be carried out using any finite difference formula.

Theorem 4.1. For \( l = 0, 1, \ldots, p-1 \), let

\[
w_{ij}^{-1} = z_{ij}^{M-1} \prod_{k=1}^{i-1}(z_{ij} - z_k), \quad r = 0, 1, \ldots, n-1,
\]

and let \( Q_i(z) \) be the balanced complex trigonometric polynomial of degree \( m \) that satisfies the interpolation conditions

\[
Q_i(z) = f_{i+j} w_{ij}, \quad r = 0, 1, \ldots, n-1,
\]

on the subset of points \( Z_i \). Then \( P(z) \), the balanced trigonometric polynomial that satisfies the interpolation conditions in (6), can be expressed in the form

\[
P(z) = \sum_{j=0}^{n-1} Q_i(z_{ij}) \prod_{k=1}^{i-1}(z - z_k).
\]

Proof. See [1].

As before, we look for a generalized barycentric formula. This formula is developed in theorem 4.2 below.

Theorem 4.2. For \( l = 0, 1, \ldots, p-1 \), let \( \hat{Q}_i(z) \) be the balanced complex trigonometric polynomial of degree \( m \) that satisfies the interpolation conditions

\[
\hat{Q}_i(z_j) = (-1)^{r-1} f_j w_{ij}, \quad r = 0, 1, \ldots, n-1,
\]

on the subset of points \( Z_i \). Then the balanced trigonometric interpolation polynomial \( P(z) \) of Theorem 4.1 can be expressed in the form

\[
P(z) = \sum_{j=0}^{n-1} (-1)^{r-1} \hat{Q}_i(z_{ij}) \prod_{k=1}^{i-1}(z - z_k).
\]

where, \( R(z) \) assumes the simple forms

\[
R(z) = \begin{cases} 1 & \text{if } p \text{ is odd} \\ \frac{z^p}{2} & \text{if } p \text{ is even} \end{cases}
\]

Proof. See [1].
Now that we have obtained the barycentric form of \( P(z) \), we can obtain that of \( T(\theta) \), the real form of \( P(z) \), very easily as follows:

\[
T(\theta) = \frac{\sum_{l=0}^{m} \ell (-1)^l U_l(\theta - \theta_i)/\sin m(\theta - \theta_i)}{\sum_{l=0}^{m} (-1)^l V_l(\theta - \theta_i)/\sin m(\theta - \theta_i)},
\]

where \( U_l(\phi) \equiv \hat{Q}_l(z) \), with \( s = e^{1/\phi} \), is the balanced trigonometric polynomial of degree \( m \) that satisfies the interpolation conditions

\[
\hat{U}_l(\theta_i) = (-1)^{l(p-1)} f_{k+lp}, \quad r = 0,1,\ldots,n-1,
\]

and \( V(\phi) \) is given by

\[
\hat{V}(\phi) = \begin{cases} 
1 & \text{if } p \text{ is odd.} \\
\cos m\phi & \text{if } p \text{ is even.}
\end{cases}
\]

We obtain a speed-up of order \( p \) both in the construction and evaluation of the polynomial as compared to the sequential FFT algorithm.

5. Chebyshev interpolation. Let \( x_j \) be \( N \) Chebyshev points

\[
x_j = \cos \theta_j, \quad \theta_j = \frac{2j + 1}{2N}, \quad j = 0,1,\ldots,N-1,
\]

in \([-1,1]\). Let \( f(z) \) be a function defined on \([-1,1]\) whose values \( f_j \equiv f(x_j), j = 0,1,\ldots,N-1 \), are given. Let \( N = np \), and consider the partition \( X = \{x_0,x_1,\ldots,x_{p-1}\} \), where

\[
X_l = \{x_{lr} = x_{l+rp}, \quad r = 0,1,\ldots,n-1\}, \quad l = 0,1,\ldots,p-1.
\]

We define a new partition, \( Y = \{y_0,y_1,\ldots,y_{q-1}\} \), with \( n = \lceil p + 1 \rceil/2 \), by

\[
Y_l = X_l \cup X_{l'}, \quad l = 0,1,\ldots,q-1, \quad l' = p - 1 - l.
\]

**Lemma 5.1.** For \( l = 0,1,\ldots,q-1 \), define

\[
\omega_{d-1} = \prod_{k\neq l,l'} (x_{lr} - x_{kr}), \quad r = 0,1,\ldots,n-1,
\]

and let \( Q_l(z) \) be the polynomial of degree \( |Y_l| - 1 \) that satisfies the interpolation conditions

\[
Q_l(z_{kr}) = f_{k+lp} w_{kr}, \quad k = l,l', \quad r = 0,1,\ldots,n-1,
\]

on the subset of points \( Y_l \). Then \( P(z) \), the interpolation polynomial on \( X \), can be expressed in the form

\[
P(z) = \sum_{l=0}^{q-1} Q_l(z) \prod_{k\neq l,l'} (z - x_{kr}). \tag{7}
\]

**Proof.** The result in (7) follows from Theorem 2.1. \( \square \)

In developing the barycentric formula we distinguish between the cases in which \( p \) is even and odd.

**Theorem 5.2.** Let \( p \) be even, and for \( l = 0,1,\ldots,q - 1 \), let \( \hat{Q}_l(z) \), be the polynomial of degree \( 2n - 1 \) that satisfies the interpolation conditions

\[
\hat{Q}_l(z_{kr}) = f_{k+lp}, \quad k = l,l', \quad r = 0,1,\ldots,n-1.
\]

Then \( P(z) \) of Lemma 5.1 can be expressed in the form

\[
P(z) = \frac{\sum_{l=0}^{q-1} (-1)^l \sin 2n\theta_l \hat{Q}_l(z)/\sin m(z_l) - T_{2n}(z_l))}{\sum_{l=0}^{q-1} (-1)^l \sin 2n\theta_l/\sin m(z_l) - T_{2n}(z_l))},
\]

where \( T_{2n}(z) \) is the Chebyshev polynomial of the first kind.

**Proof.** See [1]. \( \square \)

**Theorem 5.3.** Let \( p \) be odd, and for \( l = 0,1,\ldots,q - 2 \), let \( \hat{Q}_l(z) \), be the polynomial of degree \( 2n - 1 \) that satisfies the interpolation conditions

\[
\hat{Q}_l(z_{kr}) = \begin{cases} 
(-1)^l f_{k+lp}, & k = l, \\
(-1)^{l+1} f_{l+lp}, & k = l'.
\end{cases}
\]

Furthermore, let \( \hat{Q}_{q-1}(z) \), be the polynomial of degree \( n - 1 \) that satisfies the interpolation conditions

\[
\hat{Q}_{q-1}(z) = f_{q-1+lp}, \quad r = 0,1,\ldots,n-1.
\]

Then \( P(z) \) of Lemma 5.1 can be expressed in the form

\[
P(z) = \frac{\sum_{l=0}^{q-2} (-1)^l \sin 2n\theta_l \hat{Q}_l(z)/\sin m(z_l) - T_{2n}(z_l))}{\sum_{l=0}^{q-2} (-1)^l \sin 2n\theta_l/\sin m(z_l) - T_{2n}(z_l))} + \frac{\hat{Q}_{q-1}(z)}{T_{2n}(z_l) - T_{2n}(z_l)}.
\tag{8}
\]

**Proof.** See [1]. \( \square \)

We next show how to find the corresponding polynomials \( \hat{Q}_l(z) \), \( l = 0,1,\ldots,q - 1 \) using the FFT algorithm. We give explicit formulas for the case where \( p \) is odd. The case where \( p \) is even is solved similarly. Let \( Q(z) \) be a polynomial of degree \( m \), and let its representation in terms of the Chebyshev polynomials of order less than \( m \) be

\[
Q(z) = \frac{1}{2^{m+1}} + \sum_{k=1}^{m-1} a_k T_k(z).
\]

Rewriting the series in terms of \( z = \cos \theta \), \( z = e^{i\theta} \), we get the corresponding complex Chebyshev polynomial of degree \( m - 1 \)

\[
C(z) = \sum_{k=0}^{m-1} c_k z^k, \quad c_k = c_{-k} = \frac{1}{2} a_k, \quad k = 0,1,\ldots,m-1.
\]

Let \( Q(z) \) satisfies the interpolation conditions

\[
Q(z_j) = g_j, \quad z_j = \cos \theta_j, \quad \theta_j = \frac{2j + 1}{2m}, \quad j = 0,1,\ldots,m-1.
\]

Then \( C(z) \) satisfies the interpolation conditions

\[
C(z_j) = P(z_j) = g_j, \quad v_j = e^{i\theta_j},
\]

and vice versa. Hence, \( Q(z) \) can be obtained from \( C(z) \).

**Theorem 5.4.** Let \( \hat{C}(z) \) be the balanced complex trigonometric polynomial of degree \( n \) that satisfies the interpolation conditions

\[
\hat{C}(z_j) = \begin{cases} 
\frac{1}{2} f_{j-1+jp}, & j = 0,1,\ldots,n-1, \quad j' = n - 1 - j \quad z_j = e^{i\theta_j/2n}.
\end{cases}
\]

\[
\hat{C}(z_j) = \frac{1}{2} f_{j-1+jp}, \quad \hat{C}(z_{j'}) = \frac{1}{2} f_{j-1+j'p}.
\]

for \( j = 0,1,\ldots,n-1 \), where \( j' = n - 1 - j \) and \( x_j = e^{i\theta_j/2n} \).
Then $O(z)$, the complex Chebyshev polynomial of degree $n - 1$, corresponding to $Q_{n-1}(z)$ in (8), can be expressed in the form

$$ O(z) = \hat{O}(z/x_1^{2k}) + \hat{O}(1/(zx_1^{2k})) $$

Prove. See[1].

**Theorem 5.5.** Let $\hat{O}(z)$ be the balanced complex trigonometric polynomial of degree $n$ that satisfies the interpolation conditions

$$ \hat{O}(z_j^l) = f_j^l, \quad \hat{O}(z_j^{l+1}) = f_j^{l+1}, $$

for $j = 0, 1, \ldots, n - 1$, where $j' = n - j$, $x_1 = e^{2\pi i/2n}$ and $l < q - 1$. Then $O(z)$, the complex Chebyshev polynomial of degree $2n - 1$ corresponding to $(-2i\sin 2\theta \hat{O}(z))$ in (8), can be expressed in the form

$$ O(z) = \left( \left( \frac{2}{x_l} \right)^n - \left( \frac{-2}{x_l} \right)^n \right) \hat{O}(\frac{1}{x_l}) + \left( \left( \frac{1}{x_l} \right)^n - \left( \frac{-1}{x_l} \right)^n \right) \hat{O}(\frac{1}{x_l}) $$

Prove. See[1].

Again we obtain a speed-up of order $p$ both for the construction and evaluation of the polynomial when $N \sim n(2p), p \ll N$.

<table>
<thead>
<tr>
<th>Construction</th>
<th>Sequential</th>
<th>Parallel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluation</td>
<td>$N \log N$</td>
<td>$2n \log(2n)$</td>
</tr>
</tbody>
</table>

6. The general Hermit interpolation problem. Let $M + 1$ distinct points $X = \{x_0, x_1, \ldots, x_M\}$, in the interval $[a, b]$, be given and let $f(z)$ be a function defined on $[a, b]$, for which

$$ f_j^l = f(0)(z_j), \quad t = 0, 1, \ldots, k_j - 1, \quad j = 0, 1, \ldots, M. $$

We are interested in constructing a representation of the general Hermit interpolation polynomial $P(z)$ of degree at most $N$, $N - 1 = \sum_{j=0}^{M} k_j$, that interpolates $f(z)$ on $X$, i.e.,

$$ P^0(z_j) = f_j^l, \quad t = 0, 1, \ldots, k_j - 1, \quad j = 0, 1, \ldots, M, $$

and is most suitable for parallel computation.

Let $\{X_1, X_2, \ldots, X_p\}$ be a partition of $X$, $X = \bigcup_i X_i$, $X_i \cap X_j = \emptyset$ for $i \neq j$.

The following theorem indicates how $P(z)$ can be constructed independently and in parallel by $p$ processors, each solving a smaller general Hermit interpolation problem on one of the subsets $X_i$.

**Theorem 6.1.** For $t = 1, \ldots, p$, and $z_j \in X_i$, define

$$ q_j^l = \left( \frac{f_j^l}{t} - \sum_{s=0}^{t-1} q_j^s v_j^s \right) / v_j^0, \quad t = 0, 1, \ldots, k_j - 1, $$

where

$$ v_j^0 = \Pi_{s \notin X_i} (x_j - z_s)^n, \quad v_j^l = \frac{1}{t} \sum_{s=0}^{t-1} v_j^s z_j^{l-1-s}, $$

for $t = 1, 2, \ldots, k_j - 1$, and

$$ z_j^{l-1} = (-1)^{l-1} \sum_{x_j \in X_i} \frac{k_r}{x_j - z_j^l}. $$

Let $Q(z)$ be the polynomial of degree at most $n - 1$, $n = \sum_{x_j \in X_i} k_j$, that satisfies the following interpolation conditions.

$$ Q_j^{l}(z_j) = t q_j^l, \quad t = 0, 1, \ldots, k_j - 1. $$

Then $P(z)$, the interpolation polynomial on $X$, is given by

$$ P(z) = \sum_{i=1}^{p} Q_i(z) / \prod_{z_j \in X_i} (z - z_j)^n $$

where $R_i(z)$, like $Q_i(z)$, is a polynomial of degree at most $n_i - 1$ that satisfies the same interpolation conditions with $f_j$ replaced by 1, and $f_j^{l}$ by 0, $t = 1, \ldots, k_j - 1$, for all $j$.

Proof. As in Theorem 2.2.

We can find the formulas for $v_j^s$, $s = 0, 1, \ldots, k_j - 1$, in

$$ O(\sum_{z_j \in X_i} k_j) \leq O(n_i (X - X_i) + n_i^2) $$

operations, and the formulas for the $q_j^s$, $s = 0, 1, \ldots, k_j$, in

$$ O(\sum_{z_j \in X_i} k_j) \leq O(n_i^2) $$

operations. Let $n_i \sim n \sim N/p$, $i = 1, \ldots, p$, and assume that the $v_j^s$ are known. Each processor is then faced with a general Hermit interpolation problem of order $n_i$ that can be solved in $O(n_i^2)$ operations.

7. Conclusion. We have presented a new interpolation polynomial that is especially useful for parallel computers as its construction and evaluation requires almost no communication between the processors. The interpolation problem is divided into smaller independent subproblems that can be solved independently using any known sequential interpolation method. Thus we have reduced the problem from order $N$ to order $n \sim N/p$. Furthermore, we have developed a barycentric formula that enjoys a high degree of numerical stability as in the case with the barycentric formula for the ordinary Lagrange interpolation.

**References**

[1] [Bar-On and A. Sidi, New algorithms for polynomial and trigonometric interpolation on parallel computers, Tech. Report 806, Technion, Computer Science Department, 1990.]


A NEW DIVIDE AND CONQUER PARALLEL ALGORITHM FOR THE CHOLESKY DECOMPOSITION OF BAND MATRICES

ILAN BAR-ON
DEPARTMENT OF COMPUTER SCIENCE, TECHNION, TECHNION CITY, HAIFA 32000, ISRAEL.

Abstract.
We present a new divide and conquer parallel algorithm for finding the Cholesky decomposition of a band symmetric positive definite matrix. This is the first time such an algorithm is presented. All previously known parallel algorithms for this problem are direct implementations of the sequential methods, which as though, offer almost no speedup. Here, for the first time a parallel oriented algorithm is presented, with an approximate speedup of order p/3 given p processors. Moreover, the algorithm can be implemented on many existing parallel computers. We further discuss the more theoretical aspects of the algorithm, and show that it can be implemented in O(log m log n) time using p = (n/m)M(m)/(log log m) processors. Here, M(m) = m^β, 2 ≤ β ≤ 3, and m^θ/log m denotes the least number of processors required in order to multiply two matrices of order m in O(log m) time. This improves by a factor of log m the best previously known result for this problem.

We conclude with an application of the algorithm to the finding of the eigenvalues of a non-singular band symmetric matrix. We show by a factor of order m, is reduced to the of finding p independent decompositions of small band matrices of order n/p. Each such problem is then solved independently and in parallel. We further let AU = A, with similar notations for D_i as in (2).

Let A, U_i matrices of order m, be given by

\[ A_i = \begin{pmatrix} A_{i1} & L_{i1} \\ U_{i1} & A_{i2} \end{pmatrix}, \quad \text{where } U_i = L_i^T \text{ is upper triangular.} \]

(1)

We assume for simplicity that q = lp, where p = 2^k+1 is the number of processors. Let A', s = 0, 1, ..., k denotes a block structuring

\[ A' = \begin{pmatrix} A'_1 & L'_1 \\ U'_1 & A'_2 \end{pmatrix}, \quad \text{where } U'_i, A'_i, L'_i \in \mathbb{M}(2^n/p), \]

(2)

Here, A_i, i = 1, ..., v denotes the ith principal submatrix of order 2^n/p, i.e., rows and columns (i−1)2^n/p+1, ..., i2^n/p, of A. The off-diagonal block elements L'_i, U'_i are given by

\[ L'_i = \begin{pmatrix} 0 & 0 \\ L_{i2} & 0 \end{pmatrix}, \quad \text{and } U'_i = \begin{pmatrix} 0 \\ U_{i2} \end{pmatrix}, \]

(3)

We further let D = A, with similar notations for D'_i as in (2),

\[ D'_i = A'_i - D'_i \text{ in parallel} \]

(4)

We describe the main stages of the algorithm in the following:

For s = k, k − 1, ..., 0 do in parallel

\[ D'_i = A'_i - D'_i (D_{i+1}^{-1}) L'_{i+1}, \quad i = 2j, \quad j = 1, 2, ..., p/2^k+1. \]

(5)

Find in parallel the Cholesky decompositions,

\[ D'_i = L_i L_i^T, \quad i = 1, 2, ..., p. \]

(6)

Let,

\[ L_i = \begin{pmatrix} L_{i1} \\ U_{i1} \\ \vdots \\ U_{i,2} \\ L_{i3} \\ \vdots \\ U_{i,p-1} \\ L_{i,p} \end{pmatrix}, \]

(7)

Solve in parallel the triangular systems

\[ L_{i1} L_{i,x} = L_i, \quad i = 1, 2, ..., p-1. \]

(8)

The decomposition A = LL^T is then given by

\[ L = \begin{pmatrix} L_1 \\ L_{1,x} \\ \vdots \\ L_{p-2,x} \\ L_{p-1,x} \end{pmatrix}, \]

(9)

(10)

(11)

(12)
Theorem 2.1. Let \( N_1 \in \mathcal{M}(m), i = 0, 1, \ldots, p - 1 \) be given by
\[
(D_i^s)^{-1} E_i^s = \left( \begin{array}{c}
\vdots \\
N_i^s
\end{array} \right), \quad i = 2j + 1, j = 0, 1, \ldots, \frac{p-1}{2}, j - 1,
\]
for \( s = 0, 1, \ldots, k \). Then, given
\[\text{let,}\]
\[\pi_{i2^s} = F_i^s + H_i^s N_{2^s+1} (I - G_i^s N_{2^s+1})^{-1} E_i^s, \quad \pi_0 \equiv 0. \tag{14}\]
Proof: see[2].

Corollary 2.2. Suppose that the submatrices
\[
E_i^s, F_i^s, G_i^s, H_i^s \text{ are symmetric. Then we can implement each step of the first stage of the algorithm in } O(m^3) \text{ operations.}
\]

Step 0. We diagonalize the submatrices \( A_i^0, i = 1, 2, \ldots, p \).
Let,
\[
T_i^0 = \left( \begin{array}{ccc}
G_i^0 & V_i^0 & 0 \\
0 & \ddots & 0 \\
0 & 0 & W_i^0
\end{array} \right), \tag{17}
\]
deote the corresponding rows of \( A_0 \) after step 0.

Step \( s = 1, 2, \ldots, k \). The \( i \)-th row of \( A_i^s \) has now the form
\[
T_i^s = \left( \begin{array}{c}
\pi_{i2^s-1} \\
\pi_{i2^s} \\
\pi_{i2^s+1}
\end{array} \right), \quad i = 1, 2, \ldots, p/2,
\]
\[
T_i^s = \left( \begin{array}{ccc}
G_i^{s-1} & V_i^{s-1} & 0 \\
0 & \ddots & 0 \\
0 & 0 & W_i^{s-1}
\end{array} \right)^{-1}, \tag{18}
\]
with \( V_i^s = V_i^{(s-1)2^s} \) and \( W_i^s = W_i^{(s-1)2^s} \). We eliminate \( G_i^{s-1}, F_i^{s-1} \) in parallel by subtracting \( G_i^{s-1}(W_i^{s-1})^{-1} \) times the last row of \( T_i^{s-1} \) from the first row of \( T_i^{s-1} \), and \( F_i^{s-1}(V_i^{s-1})^{-1} \) times the first row of \( T_i^{s-1} \) from the last row of \( T_i^{s-1} \). As a result we obtain
\[
T_i^s = \left( \begin{array}{ccc}
G_i^{s-1} & V_i^{s-1} & 0 \\
0 & \ddots & 0 \\
0 & 0 & W_i^{s-1}
\end{array} \right)^{-1}, \tag{19}
\]
We then eliminate \( \pi_{s-1}^0, \pi_{s-1}^1 \) in parallel in a similar way, i.e., we subtract \( \pi_{s-1}^0 (\pi_{s-1}^1)^{-1} \) times the last row of \( T_i^{s-1} \) from the last row of \( T_i^{s-1} \), and \( \pi_{s-1}^1 (\pi_{s-1}^1)^{-1} \) times the first row of \( T_i^{s-1} \) from the first row of \( T_i^{s-1} \). The \( i \)-th row of \( A^s \) finally receives a similar form to (18), i.e.,
\[
T_i^s = \left( \begin{array}{ccc}
G_i^s & V_i^s & 0 \\
\pi_i^s & \ddots & 0 \\
0 & 0 & W_i^s
\end{array} \right). \tag{20}
\]

Theorem 3.1. Let \( T_i^s, s = 0, 1, \ldots, k \) be as in (16). Then at the end of step \( s \) of the algorithm
\[
E_i^s = (V_i^s)^{-1} E_i^s, \quad F_i^s = (W_i^s)^{-1} F_i^s, \quad s = 0, 1, \ldots, k,
\]
Step \( s = 0, 1, \ldots, k \).

Proof. The first \( s \) steps of the algorithm may be viewed as an elimination procedure applied to the linear systems
\[
A_i X_i^s = E_i^s, \quad A_i X_i^s = F_i^s, \quad i = 1, 2, \ldots, p/2,
\]
in (3) and (6). The result now follows easily from (18).

Complexity and Processor assignments. In Step 0, we assign processor number \( i = 1, 2, \ldots, p \), to the \( i \)-th row of \( A^0 \). Each processor then diagonalize its corresponding submatrix in parallel in approximately \( 2m^2/p \) operations. In Step \( s = 1, 2, \ldots, k \) the pair of processors
\[
(i-1)2^s + 1, i2^s \quad i = 1, 2, \ldots, p/2, \tag{22}
\]
is assigned to rows \( 2i - 1, 2i \) of \( A^{s-1} \). Each such pair, performs the corresponding elimination step, exchanging \( O(m^2) \) information and performing \( O(m^2) \) operations. Hence, for \( m, p \ll n \), the algorithm performs approximately
\[
2m^2/p + O(m^2 \log p) \sim 2m^2/p \text{ operations.} \tag{23}
\]

4. A pseudo code. Let \( A \) be a band s.p.d. matrix of order \( n = qm \) and bandwidth \( m \), and let \( A = LL^T \) be the Cholesky decomposition of \( A \) as in Section 2. We assume for simplicity that \( q < p \) where \( p = 2^{k+1} \) is the number of processors.

Bottom-up sweep: We compute the E's,F's,G's,II's.
Step \( s = 0 \). We compute in parallel:
\[
E_i^0 = F_i^0 = A_i^0 E_i^0, \quad \text{for } i = 1, 3, \ldots, p - 1,
\]
\[
G_i^0 = H_i^0 = A_i^0 U_i^0, \quad \text{for } i = 3, 5, \ldots, p - 1. \tag{24}
\]
Step \( s = 1, 2, \ldots, k-1 \). We perform in parallel step \( s \) of the elimination procedure as in Section 3, and then compute in parallel:
\[
E_i^s = (V_i^s)^{-1} E_i^s, \quad F_i^s = (W_i^s)^{-1} F_i^s, \quad i = 1, 3, \ldots, p/2 - 1,
\]
\[
G_i^s = (V_i^s)^{-1} G_i^s, \quad H_i^s = (W_i^s)^{-1} H_i^s, \quad i = 3, 5, \ldots, p/2 - 1. \tag{25}
\]
Step \( s = \text{odd} \). We perform step \( k \) of the elimination procedure and compute

\[
F_k^+ = (W_k^*)^{-1}F_k.
\]  
(26)

Top-down sweep: We find \( A = LL^t \).

Step \( s = k, k-1, \ldots, 0 \). We compute in parallel

\[
N_{2^k} = F_k + H_k^T N_{2^k+1} \quad \text{for} \quad i = 2j + 1, \quad j = 0, 1, \ldots, p/2^{j+1} - 1.
\]

Step \( s = -1 \). We compute in parallel

\[
A_{t+1} = N_{t+1} - U_t^T, \quad i = 1, 2, \ldots, p - 1.
\]  
(28)

Find in parallel the decompositions

\[
A_i = L_i C_i^T, \quad \text{for} \quad i = 1, 2, \ldots, p,
\]  
(29)

and finally, solve in parallel the linear systems

\[
L_i L_i^T = L_i^T \quad \text{for} \quad i = 1, 2, \ldots, p - 1.
\]  
(30)

The triangular factor \( L \) of \( A = LL^t \) is then given as in (12).

Complexity and Processor assignments. We have considered the bottom-up sweep in Section 3. In Step \( s = k, k-1, \ldots, 0 \) of the top-down sweep, processor numbers

\[
i2^k, \quad i = 2j + 1, \quad j = 0, 1, \ldots, p/2^{j+1} - 1
\]  
(31)

compute the corresponding \( N_{2^k} \), performing \( O(m^3) \) operations and exchanging \( O(m^2) \) information in parallel. Then, in Step \( s = -1 \), processor \( i = 1, 2, \ldots, p \), performs approximately \( nm^2/p \) operations. Hence, for \( m, p \ll n \), the complexity is

\[
O(m^2 \log p) + nm^2/p \cdot nm^2/p \text{ operations}, 
\]  
(32)

for a total of \( \sim 3nm^2/p \) operations. We conclude that the algorithm can be efficiently implemented on many existing parallel computers with \( S_p \approx p/3 \) speed-up, and with low communication overhead.

5. An \( O(\log m \log n) \) time algorithm. Let \( A \) be a band \( n \times n \) matrix of order \( n = gm \) and bandwidth \( m \), and let \( A = LL^t \) be the Cholesky decomposition of \( A \). We show in this section how to implement the algorithm given \( p > q/\log q \) processors. The algorithm proceeds as for the case where there are \( \min(q, p) \) processors, only now matrix operations of order \( m \) are done in parallel. We show how to implement efficiently each matrix operation such as add, multiply, invert, and find the Cholesky decomposition, in parallel. We distinguish between two cases as follows:

\[
q/\log q < p \leq qm^2/\log q: \text{We perform each step with as many processors as available. For example, with } q \text{ processors, we implement each operation in Step } 0 \text{ with one processor, and Step } s = 1, 2, \ldots, k, \text{ with } 2^s \text{ processors. Using standard parallel methods, each of the above matrix operations require at least } O(m) \text{ time. As there are only } \sim 2q \log q \text{ steps the total complexity is}
\]

\[
O(qm^2/p) + O(m \log q) = O(nm^2/p) \text{ time},
\]  
(33)

and \( S_p \approx p/3 \), as the total operation count remains the same.

\[
qm^2/\log q < p \leq qM(m)/(\log m)^2 \text{log } n: \text{We proceed as in the previous case, using only the basic matrix operations such as add, subtract, and multiply. Here, } M(1) = m^2, 2 \leq \beta \leq 3, \text{ and } m^2/\log m \text{ de... as the least number of } f \text{ processors required in order to multiply matrices of order } m \cdot O(\log m) \text{ time. For example, using the standard multiplication algorithm } \beta = 3, \text{ parallel implementation of Strassen's method gives } \beta = \log 7 \approx 2.8 \text{ see Chandra[5], and for huge } n \text{... matrices } \beta \text{ can be reduced even further, see Pan and Reif[8].}
\]

The bottom-up sweep is implemented as in [1] section 3, with complexity \( O(qM(m)/p) \). In the top-down sweep we have the following:

1. We multiply matrices of order \( m \) in \( O(\log M(m)/p) \) time, using \( p \leq M(m)/\log m \) processors.

2. Consider the inverses

\[
(I - GR)^{-1} = (I - Z)^{-1}, \quad Z = GR,
\]

in (27), where we denote for simplicity \( G_i^T \) by \( G \), and \( R_{2^{i+1}} \) by \( R \). Here, we observe from (27) that

\[
\| (I - Z)^{-1} - S_M \| \leq \sum_{i=M}^{\infty} \| Z_i \|^{1/2} \leq N/\alpha/(1 - \alpha), \quad \alpha < 1,
\]

where,

\[
S_M = \sum_{i=0}^{M-1} Z_i = \sum_{i=0}^{\log M/2} (I + Z^2)^i,
\]

and the solution converges quadratically. For example, let \( \alpha = 3^{-t/30} \) and let the precision used be \( \epsilon = 1^{-t} \). Then for \( t \geq 8 \), eleven products suffices to get an accurate inverse. The actual convergence rate is closely \( \alpha \) dependent, and even for \( \alpha = 1 - 1/m^{\alpha(1)} \) the series converges in \( O(\log m) \) steps. We will assume for simplicity that \( \alpha \) is independent of \( m \). The actual properties of this new iterative method requires further research.

3. We finally find the Cholesky decompositions of the dense \( n \times n \) matrices \( A_i^T \) in (29), using the author's new divide and conquer parallel algorithm described in [3]. A similar iterative scheme used there gives a complexity of order

\[
O(\log m)/p \text{ time using } p \leq m/(\log m)^2 \text{ processors}.
\]

As a by product we obtain the inverses of the triangular factors and the triangular systems in (30) can be solved by multiplication.

We conclude that the total complexity is

\[
O(qM(m)/p) \text{ using } p \leq qM(m)/(\log m \log n) \text{ processors},
\]

with \( S_p = O(p) \) speed-up over sequential algorithms. Furthermore, there are only \( O(\log q + \log m) = O(\log n) \) steps each dominated by the time to multiply matrices of order \( m \), an operation which can be efficiently implemented on many parallel computers.
6. An application to the QR algorithm. The QR algorithm finds the eigenvalues of A by repeatedly performing
- Find the QR factorization $A = QR$.
- Set $A = RQ$.

until the offdiagonal elements become negligible. The resulting diagonal matrix is similar to the original matrix and therefore has the same eigenvalues. Sameh and Kuck have presented a parallel QR algorithm for tridiagonal matrices see[10] which can not be applied to general band matrices. We present for the first time such an efficient and fast parallel algorithm.

Let A be a band symmetric non-singular matrix of order $n = qm$ and bandwidth m, and let $A = QR$ be the QR decomposition of A. Then it is well known, see Parlett[9], that Q is lower Hessenberg of bandwidth m, and R is upper triangular of bandwidth 2m, i.e.,

$$Q = egin{bmatrix} I & 0 & \cdots & \cdots & \cdots \\ 0 & I & \cdots & \cdots & \cdots \\ \vdots & \vdots & \ddots & \ddots & \ddots \\ \vdots & \vdots & \ddots & \ddots & \ddots \\ \vdots & \vdots & \ddots & \ddots & \ddots \\ \end{bmatrix}$$

and let

$$R = \begin{bmatrix} R_1 & S_1 & \cdots & \cdots & \cdots \\ R_2 & S_2 & \cdots & \cdots & \cdots \\ \vdots & \vdots & \ddots & \ddots & \ddots \\ \vdots & \vdots & \ddots & \ddots & \ddots \\ \vdots & \vdots & \ddots & \ddots & \ddots \\ \end{bmatrix}$$

where all submatrices are of order m. Then,

$$AL = \begin{bmatrix} E_1 & F_1 & \cdots & \cdots & \cdots \\ \cdots & \cdots & \ddots & \ddots & \ddots \\ \cdots & \cdots & \ddots & \ddots & \ddots \\ \cdots & \cdots & \ddots & \ddots & \ddots \\ \cdots & \cdots & \ddots & \ddots & \ddots \\ \end{bmatrix}$$

and therefore X can be found as follows:

$$X = L^{-1}AL = R^{-1}AR^t = Q^tR^t = RQ,$$  \hspace{1cm} (35)

is the transformed matrix sought. Let us denote X by

$$X = \begin{bmatrix} X_1 & Y_1 & \cdots & \cdots & \cdots \\ Z_1 & X_2 & Y_2 & \cdots & \cdots \\ \vdots & \vdots & \ddots & \ddots & \ddots \\ \vdots & \vdots & \ddots & \ddots & \ddots \\ \vdots & \vdots & \ddots & \ddots & \ddots \\ \end{bmatrix}$$

and let

$$Z_i = Y_i^t$$

upper triangular,

$$Z_3 = X_3$$

Now, since $A = QR$ it follows that $L = R^t$ and therefore

$$R_i Y_i = F_i,$$  \hspace{1cm} (36)

Solve in parallel the triangular systems

$$R_i X_i = E_i - S_i^{-1} Y_i,$$  \hspace{1cm} (37)

Complexity: The complexity of the algorithm is dominated by the time to find the decomposition $B = R^t R$.

7. Conclusion. We have presented a practical parallel algorithm for finding the Cholesky decomposition of a general band s.p.d. matrix that does not implement the known sequential method. We have suggested a divide and conquer approach whose inherent tree structure make it simple to implement on many parallel computers. Moreover, in most practical cases the operation count is approximately only as thrice as the sequential method and the overhead due to interprocessor communication is negligible.

Other numerical algorithms such as the conjugate gradient algorithm for solving sparse systems of linear equations, and Lanczos methods for finding the eigenvalues of symmetric sparse matrices are also sequential in nature and we believe that more parallel oriented algorithms for these problems can be found.

REFERENCES

A RECURSIVE DOUBLING ALGORITHM
FOR SOLUTION OF SOME SECOND ORDER RECURRENCES
ON HYPERCUBE MULTIPROCESSORS

AYŞE KİPER
Department of Computer Engineering,
Middle East Technical University,
06531, Ankara-Turkey

Abstract. The second-order linear recurrence formulae which results from the Fourier series coefficients of the Jacobian elliptic functions $sn^m(u,k)$, $cn^m(u,k)$, and $dn^m(u,k)$ with $m \geq 1$, are evaluated by the method of recursive doubling on an Intel iPSC/4 and performance results are discussed.

1. INTRODUCTION.

In the evaluation of functions by series approximations, the accuracy increases with the increase in the number of terms of expansions which results in an increase in computer time.

The Fourier series expansion for the twelve Jacobian elliptic functions (JEFs) have been studied and given by several authors (Abramowitz and Stegun [1], Byrd and Friedman [2], Du Val [3] Whittaker and Watson [10]. Two-term recurrence formulae have been obtained for the coefficients of these series corresponding to powers of the JEFs (Kiper [6]). The resulting recurrence formulae are of the second order and linear. Parallel evaluation of these recurrences using nested recursive product form algorithm and using method of recursive doubling on a MIMD system was considered by Kiper [6] and Kiper and Evans [7]. In this paper an implementation of the recursive doubling algorithm on an Intel iPSC/4 hypercube multiprocessor was developed for the evaluation of the mentioned recurrences and the results were discussed in terms of the speed-up and the efficiency of the parallel algorithm.

2. GENERALISATION OF RECURRENC FORMULAE OF THE FOURIER COEFFICIENTS: FOR POWERS OF THE JEFs.

The analysis of the relations for the Fourier coefficients for powers of the JEFs (Kiper [6]) shows that for a prescribed $k$ (k is the modulus of the elliptic functions) these recurrences may be represented by the common expression

$$
\omega_n^0 = a
$$

$$
\omega_n^r = \alpha(n,r)\omega_n^{(r-1)} + \beta(r)\omega_n^{(r-2)}, \quad r = 2,3, \ldots, l \quad (2.1)
$$

where $l$ is the required power and $m$ is the required number of terms in the expansion. Equation (2.1) is a second order linear recurrence relation $R(l, 2)$.

3. PREVIOUS EVALUATIONS.

A sequential evaluation of the coefficients for the Fourier expansion of the JEFs $sn^m(u,k)$, $cn^m(u,k)$ and $dn^m(u,k)$ for $m \geq 1$ were obtained and the numerical values are given for various values of $k (0.1 \leq k^2 \leq 0.9)$ (Kiper [6]). It is seen that the rate of convergence decreases as the value of $k$ and the power of the JEFs increase.

A parallel evaluation of the coefficients as formulated by Kiper [6] using the nested recursive product form algorithm in which the relation (2.1) has been expressed as the solution of a matrix system

$$\omega = A\omega + b$$

(3.1)

where

$$A = \begin{bmatrix}
0 & a(n,3) & 0 \\
\delta(4) & a(n,4) & 0 \\
0 & b(4) & a(n,4)
\end{bmatrix}
$$

$$\omega = \begin{bmatrix}
\omega_n^0 \\
\omega_n^1 \\
\omega_n^2
\end{bmatrix}, \quad b = \begin{bmatrix}
a(n,0)\omega_n^0 + b(0)\omega_n^0 \\
0 \\
\omega_n^0
\end{bmatrix}
$$

If the first $m$ terms of the expansion for a power $l$ is required, then a matrix system of size $(l - 1) \times m$ with $n = 0,1,2,\ldots,m$. The size of the system increases with the increasing required power and the number of solutions of the system increases with the increasing number of terms.

Another parallel approach to the solution of the recurrence relation (2.1) was considered by Kiper and Evans [7]. Equation (2.1) yields a form of second order linear recurrent equation which generates a vector result in two dimensions. Since, if we let

$$V_n^{(r)} = \begin{bmatrix}
\omega_n^r \\
\omega_n^{(r-1)}
\end{bmatrix} \quad \text{and} \quad A_n^{(r)} = \begin{bmatrix}
a(n,r) & \beta(r) \\
0 & 1
\end{bmatrix}
$$

(3.2)

$\quad r = 2,3,\ldots,l; \quad n = 0,1,2,\ldots,m$, then (2.1) for the prescribed power $l$ can be written as (Modi [8], Schendel [9])

$$V_n^{(l)} = A_n^{(0)} \cdot A_n^{(l-1)} \cdots A_n^{(2)} \cdot V_n^{(1)}, \quad n = 0,1,2,\ldots,m. \quad (3.3)
$$

The associative property of matrix-matrix multiplication leads us to use the recursive doubling process [9] in $O(\log l)$ steps for each $n$ ($n = 0,1,2,\ldots,m$). It must be noted that computations need $O(\ell)$ steps for each $n$ in the sequential mode. The numerical experiments of the proposed algorithm were carried out on the Sequent Balance 8000 multiprocessor with 5 processors and a comparative discussion of the results were given in [7].

4. RECURSIVE DOUBLING ALGORITHM ON HYPERCUBE MULTIPROCESSORS.

An implementation of the recursive doubling algorithm whose formulation has been already given in [7] was considered on an Intel iPSC/4 hypercube multiprocessor.
The algorithm consists of the following three steps:

Step 1: Compute \( V[r] \) locally
\[
V[r] = A[r] \ast V[r - 1], \quad r = 2, 3, \ldots, \ell
\]
- \( p \): the number of nodes
- \( n \): the problem size
- \( \ell \): \( n/p \)

Step 2: Use the recursive doubling to send and receive \( V[t] \)
for \( k = 1 \) to \( \log(p) \) do
begin
if \((\text{MyNode} < (p - 1 - 2^{k-1}))\) then
send \( V[t] \) to \((\text{MyNode} + 2^{k-1})\);
if \((\text{MyNode} > 2^k - 1))\) then
receive \( \text{tmp} \cdot V[t] \);
end;
end;

Step 3: Compute \( w[r] \)
if \((\text{MyNode} < p - 2))\) then
send \( V[t] \) to \((\text{MyNode} + 1))\);
if \((\text{MyNode} < 0))\) then
receive \( \text{tmp} \cdot V[t] \);
for \( k = 1 \) to \( \ell - 1 \) do
\[
V[k][0][0] = V[k][0][0] \ast \text{tmp} \cdot V[k][0][0] + V[k][0][1] \ast \text{tmp} \cdot V[k][1][0];
\]

The algorithm can be analysed as:

Step 1: takes \( 4(\ell - 1) \) multiplication and \( 2(\ell - 1) \) addition steps.

Step 2: takes \( 8 \ast \log(p) \) multiplication \( 4 \ast \log(p) \) addition and \( 2 \ast \log(p - 1) \) communication steps.

Step 3: takes \( 2(\ell - 1) \) multiplication, \( m - 1 \) addition and \( 2p - 3 \) communication steps.

The total number of arithmetic and communication steps involved are \((9\ell - 4 + 12 \ast \log(p))\) and \((2p - 3 \div 2 \ast \log(p - 1))\) successively.

The major advantage of the hypercube multiprocessor implementation is that the iPSC/d4 is a circuit-switched machine (contrasts to the store-and-forward counterpart) and the neighboring nodes of each node have not been found out (in order to implement the recursive doubling algorithm) as Egecioglu, Koç and Laub [4] did in their paper.

5. EXPERIMENTAL RESULTS AND CONCLUSIONS.

The effect of multiple processes was investigated by running the program on 1, 2, 3, 8 and 16 nodes \((p)\) successively and computing the total amount of time \( T_p \) (milliseconds) needed. The performance of the algorithm was measured in terms of the speed-up \((Sp)\) and the efficiency \((Ep)\). The numerical results with respect to the problem size and the number of nodes used are given in Table 1.

![Fig. 1. Speed-up](image-url)
Acknowl degment. The author acknowledges her thanks Mr. M.M. Lee at Michigan Technological University for carrying out the computations.

REFERENCES


Abstract. Using a matrix representation of a certain family of permutations, we model hypercube parallel data transmissions. Our model provides a theoretical framework for the estimation of the number of parallel transmissions involved in a given algorithm as well as the algorithms that achieve the lowest number of parallel transmissions.

1. Introduction. A d-dimensional hypercube architecture consists of $2^d$ node processors linked by the edges of a d-dimensional hypercube. The processors are numbered in a way such that the minimal number of edges between any pair of them is the number of different digits in the binary representation of their labels. Thus, two node-processors are joined if and only if their binary labels differ at exactly one bit. In such an architecture, a nonnumerical issue that is crucial to the performance of algorithms is the frequency and cost of communications among processors. These figures will depend on the underlying problem, on how the data is mapped onto the processors, and on the numerical algorithm. In this work we present a formal complexity analysis for the interprocessor communication problem associated to the product of a $2^d \times 2^d$ complex matrix times a $2^d$-dimensional vector. Our analysis assumes that all interprocessor communications are describable by the action of a certain group of permutations on the nodes of the hypercube. A matrix representation of these permutations allows us to use matrix algebra techniques to estimate the minimal number of parallel transmissions involved in a given algorithm as well as to find the algorithms that achieve that estimated lower bound. In this paper we briefly present the theoretical foundations of our communication complexity model and use some techniques derived from it to analyze the complexity and design optimal communication algorithms for computing some members of a well-known family of permutations.

2. Theoretical Background. Given $w \in N_0 = \{0, 1, 2, \ldots, \}$, we write $w = \sum_{n \geq 0} c_n 2^n$, where $c_n \in \{0, 1\}$, for each $n$. We also set $S_0 = \{ n \in N_0: c_0(n) \neq 0 \}$. This finite set is called the spectrum of $w$. The largest integer in $S_0$ is termed the degree of $w$. Given $x$ and $y \in N_0$, one can define the so-called dyadic sum, $x+y$, of $x$ and $y$ by $x+y = \sum c_n (c_n(x) - c_n(y)) 2^n$.

Proposition 1.- Let $\mathcal{N}_0$ under dyadic addition, is an abelian group. In particular, 0 is the additive identity and every element in $\mathcal{N}_0$ is its own inverse.

The initial segment of $\mathcal{N}_0$, $\mathcal{G}_d = \{0, 1, 2^2 - 1\}$ is a subgroup of $\mathcal{N}_0$ under dyadic addition. Indeed, $\mathcal{G}_d$ is the direct product of $2^{d}$ copies of $\mathcal{G}_1$, which is the additive group of integers under addition modulo 2. The Hamming weight $w(x)$, of $x \in \mathcal{G}_d$, is defined by $w(x) = \sum_n c_n(x)$.

In particular, in an initial segment of $\mathcal{G}_d$, the Hamming weight ranges through the integers between 0 and $d$. Another measure of size used in this context is the Hamming distance, $h(x,y)$, between $x$ and $y$, is defined by $h(x,y) = \sum_n (c_n(x) - c_n(y))$. Since $c_0(x,y) = c_0(x) - c_0(y)$, it is clear that $h(x,y) = |x-y|$. Some basic properties of the Hamming distance are given in the following proposition.

Proposition 2.- (a) For any $x, y, z \in \mathcal{G}_d$, $h(x+y, z) = h(x, y) + h(x, z)$.
(b) If $S_0 \cap \{1, 2, \ldots, \}$ = 0, then $h(x+y, z) = h(x, z)$.
(c) If $d > \max\{\deg w, \deg z\}$, then, for any $x, y, z \in \mathcal{G}_d$,
$$h(2^d x + 2^d y + w) = h(x, y) + h(x, w).$$

Note that if $x$ and $y$ in $\mathcal{G}_d$ are complementary, i.e., $c_n(x) \neq c_n(y)$, then $h(x, y) = d$. This, in fact, is the necessary and sufficient condition for two integers in $\mathcal{G}_d$ to be complementary.

A permutation $\sigma : \mathcal{G}_d \rightarrow \mathcal{G}_d$ will be called a nearest neighbor transposition (NNT) permutation, if $h(\sigma(x), y) = 1$, $x \in \mathcal{G}_d$. Thus, such permutations modify a single bit in the binary representation of each $x \in \mathcal{G}_d$. The NNT permutation will be called perfect if the bit modified is always the same. For instance, if $y = z \oplus 1$, then the NNT permutation $\sigma(y) = z \oplus 1$ and $\sigma(z) = x \oplus 2$ are perfect NNT permutations in $\mathcal{G}_d$. In general, it is clear that for each $d \geq 1$, there are $d$ perfect NNT permutations on $\mathcal{G}_d$ and they are of the form $\sigma(y) = y \oplus 2^j, x \in \mathcal{G}_d, j = 0, 1, \ldots, d - 1$. Let $C$ be the set of all complex numbers, $\mathbb{C}$, and $GL_2(\mathbb{C})$ the linear space of all $2^d \times 2^d$ complex matrices and $GL_2(\mathbb{C})$ the linear space of non-singular matrices with complex entries. In order to get a matrix representation for the permutations we first define the matrix representation of $\mathcal{G}_d$, $R : \mathcal{G}_d \rightarrow M_2$, as
$$R(0) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I_2 \text{ and } R(1) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = J_2.$$

Let $\otimes$ denote the tensor (also Kronecker or direct) product of matrices.

Since $GL_2$ is the direct product of $d$ copies of $\mathcal{G}_d$, the map $K(\cdot, d) : x \in GL_2 \rightarrow K(x, d) \in GL_2(\mathbb{C})$, where
$$K(x, d) = \Lambda^d(C_2, \otimes) \otimes \otimes \otimes \otimes \otimes$$

is a matrix representation of $GL_2$. The $d$ perfect NNT permutations on $\mathcal{G}_d$ are represented by the matrices $K(\mathcal{G}_d, d, j, 0, 1, \ldots, d - 1)$. In general, being $K(\cdot, d)$ a representation, $K(x+y, d) = K(x, d)K(y, d)$ and the set $U_{2^d} = \{ K(x, d) : x \in \mathcal{G}_d \}$ is a commutative subgroup of $GL_2(\mathbb{C})$. Furthermore, $K(x, d)$ is its own inverse.

If $d = 2$, the elements in $U_{2^2}$ are
$$K(0, 2) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, K(1, 2) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \text{ and } K(2, 2) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

Proposition 3.- Let $\lambda = (\lambda_0, \lambda_1, \ldots, \lambda_{d-1})$ and let
$$D(\lambda) = \begin{pmatrix} \lambda_0 & \cdots & \lambda_{d-1} \\ \vdots & \ddots & \vdots \\ \lambda_{d-1} & \cdots & \lambda_0 \end{pmatrix}.$$
Let $A, B \in M_{2^k}$. Then, for any $x \in G_{2^k}$
\[
D(x, AB) = \sum_{y \in G_{2^k}} D(x, A)D(y, B)
\]
where $B_2 = K(x, d)KD(x, d)$.

The following quantity will be useful in the estimation of the number of par-
allel transmissions involved in computing with $A$. Given any $A \in M_{2^k}(C)$, $A \neq [0]$, we define $\delta(A) = \max\{w(v) : D(x, A) \neq [0]\}$. We call this quantity the Hamming diameter of $A$.

**Proposition 5.** (a) Given any $A \in M_{2^k}$, $B \in M_{2^k}$, $\delta(A \otimes B) = \delta(A) + \delta(B)$.
(b) Given any $A, B \in M_{2^k}$, then $\delta(AB) \leq \min\{\delta(A), \delta(B)\}$.

This inequality can be strict even if both matrices are different from the zero
matrix.

2. Hypercube Parallel Transmissions. Our hypercube machine consists of $2^n$ node processors. Each processor is endowed with a certain number of vector registers of length $2^k$. Given a vector register $v_i$ in processor $P_i$ and a vector register $v_j$ in processor $P_j$, it will be always possible to send the content of the $j$-th component of $v_j$ to the $i$-th component of $v_i$. Furthermore, simultaneous exchanges of the contents of the $j$-th components of any two vector registers $v_i$ and $v_j$ is allowed. Thus, if we assume that at a certain stage of a computational process, the $2^{n+k}$-dimensional data vector $c$ is loaded as a two-dimensional array: $[c(i, j)]_{i,j \in G_{2^k}}$, where $i$ and $j$ are the processor's and vector register component's labels respectively, all possible interprocessor data movements are representable by a set of permutations on $G_{2^k}$. Each of these permutations acts on the segment $c_j = \{c(0, j), c(1, j), \ldots, c(2^k - 1, j)\}$ of $c_j$ and therefore the whole action of the permutations $P_j$ corresponds to the action of the direct sum $P = \bigoplus_j \mathcal{G}_{2^k}P_j$ on $c$, antilexicographically ordered. The permutation $P$ will be called an hypercube parallel communication (HPC). By representing each $P_j$ in terms of our formulas in section 2,
\[
P = \bigoplus_j \mathcal{G}_{2^k}(\sum_{x \in G_{2^k}} D(x, P_j)K(x, d))
\]

An hypercube parallel transmission (HPT) is now defined as the special type of HPC where $P_j = D_jK(2^k, d)$, here $D_j$ is a diagonal matrix with 1's and 0's on its diagonal and $0 \leq j < 2^k$. If $j = r$ for all $r$, then
\[
P = \bigoplus_j \mathcal{G}_{2^k}(I_{2^k} \otimes K(x, d)).
\]

Any HPC is performed through a sequence of HPT's. A lower bound for the number of HPT's involved in an hypercube parallel communication is $\max \{\delta(P_j) : j \in G_{2^k}\}$. This lower bound, however, is not always achievable.

**Theorem 3.** Any permutation on a $2^k$-dimensional data vector is computeable with a minimum number of HPTs.

Proof: Let $A$ be a matrix representation of a permutation on $C_{2^k}$. Then $A = \sum_{x \in G_{2^k}} D(x, A)K(x, d)$. Let $\text{Im}D(x, A)$ and $\text{Im}D(x, A')$ be the image subspaces of $D(x, A)$ and $D(x, A')$ respectively. Since $A$ is a permutation,
\[
C_{2^k} = \bigoplus_{x \in G_{2^k}} \text{Im}D(x, A) = \bigoplus_{x \in G_{2^k}} \text{Im}D(x, A')
\]
and the permutation $A$ can be written as the direct sum $A = \bigoplus_{x \in G_{2^k}} T_x$, where $T_x$ is the restriction of $K(x, d)$ to $\text{Im}D(x, A')$. Now, the minimum number of HPTs required by $A$ is equal to the maximum $\delta(T_x) : x \in G_{2^k}$, which is, in fact, equal to $\delta(A)$.

**Theorem 4.** Let $A \in M_{2^k}$, be such that $A = B \otimes C$, with $B \in M_{2^k}$ and $C \in M_{2^k}$.

Let's also assume that computing with $B$ requires $\delta(B)$ HPTs. Then computing with $A$ also requires $\delta(B)$ HPTs.

Proof: Since $A = B \otimes C = (B \otimes I_2)(I_2 \otimes C)$, the process of computing on $z = Au$ can be made in two steps, $(1) y = (I_2 \otimes C)u$ and $(2) z = (B \otimes I_2)y$. To implement step (1) we divide $u$ into $2^k$ segments $u_x$, $x \in G_{2^k}$, each of length $2^k$. Then we load $u_x$ in processor $x$ and compute $y_x = Cx$. This computation requires no interprocessor communications. As for step (2), according to Proposition 6, $(1) \delta(B \otimes I_2) = \delta(B) + 4|\delta(I_2)| = \delta(B)$.

Theorem 4 establishes that a significant reduction in the number of HPTs involved in computing with $A$ is achievable whenever $A$ supports a tensor product factorization. This result explains, for instance, the good communication properties shown by the Cooley-Tucker FFT.

Besides estimating the HPT complexity of any algorithm expressible in
matrix language, our model helps in finding the best interprocessor data flow
algorithms. We will consider, by way of example, algorithms computing
certain permutations. First, we need to set up some corollaries of Proposition 5.

**Corollary 2.** If $\sigma : G_{2^k} \rightarrow G_{2^k}$ is a permutation such that $\sigma^2 = id$ and if we write $D(\sigma^2) = D(\sigma_0^2, \sigma_1^2, \ldots, \sigma_{2^k-1}^2) = D(x, P_c)$ where $P_c$ is a matrix representation of $\sigma$, then $\text{K}(y, d)^{\sigma} = \text{K}(x, d)^{\sigma}(y, d)^{\sigma}$. Definition. Let $A \in M_{2^k}$. We define $\Sigma_A = \{x \in G_{2^k} : D(x, A) \neq [0]\}$.

**Corollary 2.** If $\sigma : G_{2^k} \rightarrow G_{2^k}$, then $\{x \in G_{2^k} : D(x, A) \neq [0]\}$. (b) For any $x \in G_{2^k}$, there is one and only one $\sigma \in \Sigma_A$ such that $\sigma^2 = 1$. Now, we apply our theoretical framework to the analysis and design of parallel algorithms for computing well known permutations such as the bit-reversal and the index digit permutations. For instance, let's consider $\beta_x : G_{2^k} \rightarrow G_{2^k}$ the 3-digit bit-reversal permutation. Then $\beta_x$ is determined by the transpositions $(1, 4)$, $(3, 6)$. Now, for $i = 0, 1, 2, 3$, $2i+\beta_x(2i) = 0$ and $2i+1 + \beta_x(2i+1) = 5$. Thus $D(0, F_{2^k}) = D(1, 0, 1, 0, 1, 0, 1)$ and $D(5, F_{2^k}) = D(0, 1, 0, 1, 0, 1, 0)$. Therefore, $\beta_x$ is reduced to $K(3, 5)$ acting on the space $\text{Im}D(5, F_{2^k})$. Now, since $w(5) = 2$, $\delta(K(3, 5)) = 2$ and therefore the HPT complexity of computing $\beta_x$ is 2. An algorithm achieving that complexity is obtained by simply factorizing $K(3, 5) = K(1, 3)K(1, 3)$. Our second example starts with the permutation $\sigma$ determined by the cycle $(0, 123)$. Such a cycle induces a permutation $\sigma : G_{2^k} \rightarrow G_{2^k}$. This permutation is defined as $\sigma(c_0(z)^2 + c_2(z)^2 + c_1(z)^2 + c_0(z)) = c_1(z)^2 + c_2(z)^2 + c_1(z)^2 + c_0(z)$ and sometimes termed index-digit permutation. A direct calculation shows that $\Sigma_{\sigma} = \{0, 3, 5, 6, 9, 12, 15\}$. Since $\delta(w(15)) = 4$ and $\delta(P_{2^k}) = 4$. The algorithm performing $P_{2^k}$ in four HPTs is obtained by restricting the actions of $(z, d)$ to $\text{Im}D(z, P_{2^k})$, for $x \in \Sigma_{\sigma}$. For $x \in \Sigma_{\sigma}$, $x \neq 15$, the operator $K(x, d)$ factors as a product of two perfect NNT permutations. However, since $w(15) = 4$, $K(15, 4)$ will factor as a product of four perfect NNTs.

**References**


Doping-induced anchoring transitions at liquid crystal surfaces

P. I. C. Teixeira and T. J. Sluckin

Faculty of Mathematical Studies, University of Southampton
Southampton SO9 5NH, United Kingdom

1. INTRODUCTION

If a nematic liquid crystal (LC) is spread on top of an anisotropic substrate, the energy of the LC molecules at the nematic-substrate interface will depend on orientation, thus inducing preferential alignment of the nematic director.1,2,3 This phenomenon, known as anchoring, plays a crucial role in the fabrication of LC display devices, which rely on a delicate control of anchoring at LC-polymer interfaces.4 It is known that the anisotropic anchoring forces depend on the nature of the substrate,5-7 which can be either a solid or a liquid.8

Recently, it was shown that discontinuous changes in anchoring direction - anchoring transitions - can occur as a result of changing concentrations of one or more adsorbrates.4 We are concerned with modelling the anchoring transition observed by Pieranski et al.4,6 which is driven by changes in the amount of water vapour adsorbed on a gypsum substrate. This is a transition between two monostable planar anchorings, which is notoriously difficult to study theoretically. We therefore started by considering a simpler case, that of a transition between homeotropic and planar anchorings.

2. THEORY

Following Sullivan and co-workers,7,9,11 we used a mean-field approximation to the Helmholtz free energy functional of a non-uniform nematic liquid to derive a simpler, Landau-de Gennes free energy functional. The main advantage of this method is that we obtain explicit expressions for the phenomenological coefficients appearing in Landau-de Gennes theory in terms of the intermolecular (and surface) potentials.

Besides mean-field, the basic approximation is the assumption of a step-function variation of the density and order parameter profiles. Although this is clearly incorrect, as it neglects the role of surface adsorption,10 we expect the resulting theory to provide at least qualitative insight into the mechanism determining the equilibrium alignment.

We first generalised Sullivan and co-workers' approach to consider a single-component uniform nematic phase at a (plane) surface. Neglecting biaxiality, the surface term in the expression of the surface tension is,

\[ f_s = \omega_c + \omega_2 \cos \beta + \omega_4 \cos^2 \beta + \omega_6 \cos^4 \beta \]

where \( \beta \) is the tilt angle, i.e. the angle between the normal to the surface, \( \mathbf{n} \), and the nematic director, \( \mathbf{a} \). The \( \omega_2 \) and \( \omega_4 \) are the nth-order Legendre polynomials.

We can estimate the orientation favoured by the surface by minimising only \( f_s \) instead of the full surface tension.10,11

\[ \text{Equation (11) truncated after the } P_4 \text{ term is known in the literature as the Rapini-Papoular form of the surface anchoring energy,} \]

which has been extensively used in both theoretical and experimental work.

This theory is straightforwardly generalised to a binary LC mixture at a surface. \( f_s \) is still given by (1) and the \( \omega_n \) are linear combinations of \( \psi_n \psi_m \), the third moments of the coefficients in the spherical harmonic expansion of the intermolecular potential between species \( i \) and \( f_{ij} \) for \( (1,1) \) = (000), (220), (202), (222) and (224) and \( V_{ext} \), the integrated LC-surface interaction. In particular, \( \omega_4 \) is a function of \( V(224) \) only. The coefficients in the linear combinations are products of the total density and of the concentrations and order parameters of the two components.

Minimization of \( f_s \) with respect to \( \psi \) yields the phase diagram shown in fig. 1.

![Figure 1: The anchoring phase diagram of a LC in \( w_2, w_4 \) space.](image)

As \( w_2 \) and \( w_4 \) change as functions of temperature and concentration, the trajectory of a system in \( w_2, w_4 \) space may cross one or more of the boundaries between different anchoring domains, at which point an anchoring transition occurs which is one of the following: first- or second-order if \( w_4 \) < 0 or second-order if \( w_4 \) > 0. Sullivan and co-workers7,11 argued that the \( V(224) \) term in the interaction potential \( f_s \) coming from quadrupole-quadrupole interactions and short-range anisotropic repulsive and attractive forces are essentially positive, leading to non-negative \( w_2 \) and second-order transitions (however, \( w_4 \) may vanish), as we shall see.
In order to calculate $w_n$ and $\psi$ we need a specific model for $V(l, l', l'')$. In a first approach, we used the simple Telo da Gama model of LCs, which includes (000), (220), (022) and (002) terms, augmented with a (224) term coming from quadrupolar interactions and an inverse-power law surface potential. Note that, in this simple mean-field theory, only $(l, l', l'')$ terms with $l = 0$ contribute to the free energy of the bulk phases and therefore, given our step-function approximation for the density, concentration and order parameter profiles, the addition of (202) or (224) terms will not change the density, concentration and order parameter profiles, and therefore, given our step-function approximation for the order parameter.

The addition of (202) or (224) terms will not change the surface free energy of the pure components by the Lorentz-Berthelot rule (this is exact in the case of quadrupolar interactions, which allows for the vanishing of $w_n$ if the quadrupole moments of the two components have the same magnitude but opposite signs).

3. RESULTS

Fig. 2 shows a cross-section (taken at constant reduced quadrupole moment $Q/\sqrt{G\sigma^5}$) of the $(x_2, Q^*)$ phase diagram of a binary LC mixture whose components differ by i) the sign (but not the strength) of their quadrupole moments, and ii) the fact that the surface favours homeotropic alignment of component 1 ($V^1_{\text{ext}} < 0$) and planar alignment of component 2 ($V^2_{\text{ext}} > 0$). $x_2$ is the concentration of component 2. $B^*_V$ is a normalised combination of $V_{\text{ext}}$ and $V(l, l', l'')$.

![Figure 2: A cross-section of the $(x_2, Q^*)$ phase diagram of the binary LC mixture described in the text for $Q^* = 1.0$. $B^*_V$ is the strength of the surface potential.](image)

As $Q^* = Q^*_1 - Q^*_2$ increases, re-entrant conical anchoring becomes possible. In fig. 3 we plot the anchoring angle versus $x_2$ along trajectories I and II in fig. 2. The latter clearly displays anomalous, or re-entrant, behaviour.

![Figure 3: Tilt angle vs. $x_2$ along trajectories I and II in fig. 2.](image)

4. CONCLUSIONS

We used a simple mean-field theory to derive a Landau-de Gennes expression for the surface free energy of a mixture of nematic LCs in contact with an anisotropic substrate. In spite of the fact that it contains a number of oversimplifications, the theory predicts fairly rich and interesting anchoring behaviour, including re-entrant conical anchoring, as a function of composition, when applied to a simple LC model.

REFERENCES

Abstract - Using algebraic processors general properties of SO(3) - invariant free energy expansion of biaxial liquid crystals are studied. This is achieved by decomposing invariants in terms of integrity bases as proposed by Judd et.al., Gaskell et.al., and Bistricky et.al. With the help of this approach a rigorous continuum free energy of general biaxial nematics is analyzed as an expansion in components of a symmetric and traceless tensor order parameter field Q_{\alpha\beta} and its derivatives \partial Q_{\alpha\beta}. Next, a general theory of flexopolarization and a classification of local, polar structures in biaxial systems is offered. Finally, some consequences for biaxial smectic systems are summarized.

I. INTRODUCTION

Algebraic processors, like Macsyma or Mathematica, have been designed to perform symbolic, numerical and graphical calculations easily and to arbitrary precision. One can also use these processors to prepare input for, or analyse output from other external programs. Below we would like to summarize the results for continuum theory of biaxial liquid crystals obtained by combining symbolic and numerical possibilities of Maceyma.

A. Order Parameters

Orientational properties of liquid crystals are described in terms of irreducible spherical tensor fields Q^{(L)}(r) of angular momentum L and of components Q^{(L)}_m(r) [1]. Out of them the most important is \nu=2 quadrupole tensor which describes anisotropic part of electric- or magnetic susceptibilities. In cartesian representation Q^{(2)} is identified with a second - order, symmetric and traceless tensor field Q(r) of components Q_{\alpha\beta}(r). For polar liquid crystals additionally Q^{(1)}_m(r) field must be retained.

B. Integrity bases

Theoretical studies of physical properties of the systems described in terms of Q^{(L)} are based on the nonequilibrium free energy density expansion around an isotropic phase i.e. Q^{(L)} = 0. This expansion is an SO(3) symmetric polynomial in the components of irreducible spherical tensors Q^{(L)} and their derivatives. Consequently, an important problem to solve is that of determining all SO(3) invariants which are homogeneous polynomials in components Q^{(L)}_m and derivatives \partial Q^{(L)}_m. One can prove that for a finite set of Q fields it is possible to construct a finite basis of invariant polynomials (integrity basis) such that all other invariants can be written as polynomials of these basic invariants. This very elegant group theoretical method has been applied to problems with SO(3) symmetry by Judd et.al.[2], Gaskell et.al.[3] and Bistricky et.al.[4].

For large L, calculations of the integrity basis elements and studies of their properties are nontrivial algebraic problems, which appear to be perfectly suited for algebraic processors [5-7]. Here we summarize some of the results, obtained for the L = 2 tensor field and for the L = 1 vector field. Some other examples are found in refs.[2-7].

II. ELASTIC AND FLEXOPOLARIZATION MODES OF BIAXIAL LIQUID CRYSTALS

Elastic free energy of biaxial liquid crystals is defined as an expansion in Q_{\alpha\beta} and \partial Q_{\alpha\beta}, where only first and second order terms in derivatives \partial Q_{\alpha\beta} are retained. Thus, the expansion contains SO(3) - symmetric invariants built up from the tensors Q_{\alpha\beta}Q_{\gamma\delta}\ldotsQ_{\rho\sigma}(Q_{\mu\nu},\eta) and Q_{\alpha\beta}Q_{\gamma\delta}\ldotsQ_{\rho\sigma}(Q_{\mu\nu},\eta)(Q_{\xi\tau},\zeta) obtained by means of contractions with the Kronnecker deltas and the Levi-Civita tensors.

Similarly, the deformation induced polarization of biaxial systems depends on Q_{\alpha\beta} and Q_{\alpha\gamma}, \nu at each point. Since the effect is linear in deformations the corresponding flexopolarization part of the free energy must include the class of all linearly independent SO(3) - symmetric invariants P_\xi Q_{\alpha\beta}Q_{\gamma\delta}\ldotsQ_{\rho\sigma}(Q_{\mu\nu},\eta), where P_\xi is the polarization field (P_\xi = P^{(1)} = Q^{(1)}).

Now, decomposing the invariants of Q_{\alpha\beta}, Q_{\alpha\beta}, \nu and P_\xi in the corresponding integrity basis, one finds that the most general free energy expansion to all powers of Q_{\alpha\beta} and up
to second order in $Q^{a\beta\gamma}$ contains 39 basic elastic modes [6], where three of them are chiral. These generalize the concept of the splay-, bend-, and twist deformations of the director field, introduced by Oseen, Zocher and Frank. The associated, temperature dependent elastic constants are analytical functions of $TrQ^2$ and $TrQ^3$.

Similarly, the general flexopolarization free-energy density of chiral biaxial liquid crystals is composed of 12 basic deformation modes [7]. These are multiplied by arbitrary polynomials in $TrQ^2$ and $TrQ^3$ which define temperature- and position dependent flexocoefficients.

In both cases simpler forms of the free energy densities are obtained from the general expansion by imposing additional symmetry restrictions on the field $Q$. Details are given in refs. [6,7].

III. POLAR STATES OF BIAXIAL LIQUID CRYSTALS

With the help of integrity basis one finds very convenient approach to study selection rules for broken symmetry states in an arbitrary Landau free energy expansion [7]. As an example we investigate the correlation between polar nematic states and the properties of the integrity basis for invariants composed of the components of $p^{(1)}$ and $Q^{(2)}$.

The integrity basis for invariants of two order parameters $p^{(1)}$ and $Q^{(2)}$ is composed of six invariants $I_{a\beta}$ whose degrees of $p^{(1)}$ and $Q^{(2)}$ are $\alpha$ and $\beta$, respectively. The invariants in Cartesian representation can unambiguously be identified as [7]:

\[
\begin{align*}
I_{02} &= TrQ^2, \\
I_{03} &= TrQ^3, \\
I_{20} &= P_\alpha P_\alpha, \\
I_{21} &= P_\alpha Q_\alpha P_\beta, \\
I_{22} &= P_\alpha Q_\alpha P_\beta - \frac{1}{3} Tr(Q^2) P_\alpha P_\alpha, \\
\text{and} \\
I_{33} &= P_\alpha P_\beta P_\gamma c_{\alpha\mu\nu} Q_{\mu\nu} Q_\nu
\end{align*}
\]

where

\[
\begin{align*}
108 (I_{33})^2 &= -54 I_{02} I_{20} (I_{22})^2 + 54 I_{02} (I_{21})^2 I_{22} - 9 (I_{02})^2 (I_{21})^2 I_{20} + 2 (I_{02})^3 (I_{20})^3 - 36 I_{03} (I_{21})^3 + 108 I_{02} I_{21} I_{22} I_{20} \\
&- 12 (I_{03})^2 (I_{20})^3 - 108 (I_{22})^3.
\end{align*}
\]

Consequently, the Landau free energy expansion for polar nematics is a stable polynomial in $I_{a\beta}$ and, at most, a linear function in $I_{33}$. One finds that in addition to the uniaxial and biaxial ferroelectric nematic phases which are generated by the first five invariants, there exists a biaxial chiral ferroelectric nematic phase, generated by the $I_{33}$ term.

From the form of $I_{33}$ it is clear that the following must be fulfilled for the possible existence of the chiral biaxial phase: i) chiral molecules with a large dipole moment component, perpendicular to the long molecular axis ii) large molecular biaxiality, probably of the same order as the one observed in thermotropic biaxial nematics.

Similar statements hold for the smectic-C* phase. One finds that if the $S_C^*$ phase is stabilized due to the piezoelectric coupling between $P$ and a density wave then it must be described as a biaxial, uniform spiral with at least two nonvanishing commensurate harmonics. Since the polarization in the $S_C^*$ phase is perpendicular to the local director, additionally the biaxial piezoelectric coupling invariant $I_{33}$ must vanish.

For nonzero value of the $I_{33}$ invariant another phase with the $S_C^*$ symmetry may be more stable. In this phase, the polarization is not perpendicular to the local director. Intrinsic biaxiality of chiral liquid crystalline molecules is the driving force, stabilizing this phase.

The above predictions are in accordance with recent expectations that the non centrosymmetric biaxial molecules with negative dielectric anisotropy may be good candidates to form phases with local ferroelectric, biaxial nematic order.

A thorough discussion of the phase diagrams for the cases discussed above will be presented elsewhere.

REFERENCES

1. ERICKSEN-LESLIE EQUATIONS

We consider simple shear flow between parallel plates at a distance $2h$ apart which are parallel to the $x$-$y$ plane. We assume that the upper plate at $z = h$ is at rest while the lower one at $z = -h$ moves with velocity $V$ in the $y$ direction. The state of the nematic liquid crystal is described by its velocity $v(x, y, z, t)$ and its anisotropic axis $n = n(x, y, z, t)$ where $|n| = 1$.

We first investigate simple shear flows of the form
\[ v = (0, v(z, t), 0) \quad n = (0, \cos \theta(z, t), \sin \theta(z, t)). \] (1.1)

For flows of the form (1.1) the Ericksen-Leslie equations are
\[ \frac{\partial v}{\partial t} = \frac{\partial}{\partial z} \left( g(\theta) \frac{\partial v}{\partial z} + m(\theta) \frac{\partial \theta}{\partial z} \right) \quad -h \leq z \leq h \] (1.2)
\[ 2\gamma_1 \frac{\partial \theta}{\partial t} = 2f(\theta) \frac{\partial^2 \theta}{\partial z^2} + \frac{\partial f}{\partial \theta} \left( \frac{\partial \theta}{\partial z} \right)^2 - 2m(\theta) \frac{\partial v}{\partial z} \] (1.3)

where
\[ g(\theta) = \alpha_1 \sin^2 \theta \cos^2 \theta + \frac{\alpha_3 - \alpha_2}{2} \sin^2 \theta + \frac{\alpha_6 + \alpha_5}{2} \cos^2 \theta + \frac{\alpha_4}{2}; \]
\[ m(\theta) = (\gamma_1 + \gamma_2 \cos 2\theta)/2; \]
\[ f(\theta) = \kappa_1 \cos^2 \theta + \kappa_2 \sin^2 \theta; \]
\[ \rho \] is the density; $\alpha_1, \ldots, \alpha_6$ are the Leslie viscosities; $\gamma_1, \gamma_2, \kappa_1, \kappa_2$ are the Frank elastic constants; and $\gamma_1 = \alpha_3 - \alpha_2, \gamma_2 = \alpha_6 - \alpha_5$. Thermodynamic inequalities imply that $g(\theta) > 0$ and $f(\theta) > 0$ for all $\theta$ and that $\gamma_1 > 0$ [4]. We shall only consider flows in the non-flow-orienting regime $\gamma_1 > 1/2$, so $m(\theta) > 0$ for all $\theta$.

We utilize the “strong anchoring” condition for $n$, i.e.,
\[ \theta(-h, t) = \theta(h, t) = \theta_p. \] (1.4)

where $\theta_p = 0$ or $\pi/2$ and the “no-slip” boundary condition for $v$
\[ v(-h, t) = 0, \quad v(h, t) = 0. \] (1.5)

For steady flow,
\[ g(\theta) \frac{\partial v}{\partial z} = c \]

where $c$ is an integrating constant for (1.2), and (1.6) can be used to eliminate $\partial \theta/\partial z$ from (1.3) to obtain
\[ 2f(\theta) \frac{\partial^2 \theta}{\partial z^2} + \frac{\partial f(\theta)}{\partial \theta} \left( \frac{\partial \theta}{\partial z} \right)^2 - 2m(\theta) \frac{\partial v}{\partial z} = 0, \]
\[ \theta(-h) = \theta(h) = \theta_p. \]
1. At $\epsilon = .5$ the out-of-plane instability occurs first. At $\epsilon = .86$, the in-plane instability occurs first.

Fig. 1.

3. COMPUTATIONAL RESULTS

Computational results for solution branches were obtained using the software package AUTO [3]. Our computational results show that the first instability can be in-plane or out-of-plane, depending on the material constants [5]. We model the behavior of 8CB just above the smectic A-nematic transition temperature with $\theta_0 = \pi/2$ by the following material constants for $\epsilon > 0$:

$$\alpha_1 = 12|\alpha_2|, \quad \alpha_2 = -7, \quad \alpha_3 = \epsilon|\alpha_2|,$$
$$\alpha_4 = .58, \quad \alpha_5 = 7, \quad \alpha_6 = \epsilon|\alpha_2|,$$
$$\kappa_1 = 1.41 \times 10^6, \quad \kappa_2 = 1.023\kappa_1, \quad \kappa_3 = 2.605\kappa_1, \quad \rho = 1,$$

where viscosities are in poise, elastic constants are in dyne, density is in g/cm$^3$, and $h = 1$ cm. In Figure 1 above, the first in-plane instability occurs at the first turning point and the position of the first out-of-plane instability is marked by "+.

Next, we present computational results for the out-of-plane solution branch of the form (2.7) which has been continued from the bifurcation point for 8GB at 35° C [5]. These computations used the boundary data $\theta(-h) = \theta(h) = 0$, $\phi(-h) = \phi(h) = \pi/2$. The position of the bifurcation point in Figure 2 is marked by "+.

Fig. 2.

The profiles of solution $u(z)$, $\phi(z)$, and $\theta(z)$ are given in Figure 3 at the points on the solution branch marked by A–W.

Fig. 3.

REFERENCES

Abstract - We present the results of an experimental study of some hydrodynamic instabilities found in a large aspect ratio nematic liquid crystal device which is subjected to a linear oscillatory shear. The instabilities depend on shear rate, frequency and gap width of the cell. They also fall into two different basic classes for thick and thin cells.

1. INTRODUCTION

Hydrodynamic instabilities in thin homogeneously aligned nematic liquid crystals which are subjected to linear oscillatory shear were first investigated by Clark, Saunders, Shanks and Leslie (1981). They found an instability in the form of 1 cm wide bands aligned in the direction of the shear which have alternating in-phase and out-phase twist separated by their dark regions. The layers were approximately 10μm thick and the frequency range was 10^-100Hz. Thus the bands are large scale effects as the width of each is approximately 1000 layer thicknesses. In addition, Clark et al carried out a stability analysis of the equations of motion and obtained good agreement between theory and experiment. One crucial assumption in their analysis is that the critical shear rate for the appearance of the instability is independent of frequency. This appeared to be borne out by their observations for thin layers.

Clark et al also observed the formation of mechanical Williams domains for shear rates above the first instability described above. These are the mechanical equivalent of the Williams domains found in electrohydrodynamic convection (see for example Blinov (1983)). They have length scales of the order of twice the layer thickness and are thus microstructures when compared with the bands described above. The mechanical Williams domains have recently been investigated by Kochevnikov (1986) for normally orientated nematics with applied oscillatory shear frequencies in the range $10^2 - 10^5$ Hz. Finally, Guazzelli (1990) has also observed mechanical Williams domains when an elliptical oscillatory shear is applied to thin layers of nematics.

In the present study we have extended the work of Clark et al to investigate the effects of layer thickness on the observed instabilities. We have found that the critical shear rates for the first appearance of the large scale instabilities are strongly dependent on frequency although there is some evidence for independence at high frequencies. Patches of Williams domains are the first instabilities to appear with increase of shear rate for layers thicker than ~25 μm whereas the banded structures arise first for thin layers in agreement with the results of Clark et al. Finally, a nonlinear interaction between the mean flow field and the Williams domains can re-orientate the rolls so that there is both a change in direction and length scale of the microstructure. This in turn is accompanied with a novel long term 'memory effect' in the device.

2. EXPERIMENT

The apparatus consisted of two optically flat (1/5) glass blocks of dimensions 10x8x1 cm mounted on an IVAR frame and spaced using three micrometers which are accurate to ±0.5 μm. The lower block is mounted on linear bearings and is connected to a large electromagnetic vibrator. The amplitude of the applied vibration is measured both by a linear displacement device and using the Michelson interferometer technique suggested by Ben-Yosef, Ginio and Wetz (1974).

The nematic liquid crystal used was 1085 TNC which has a positive coefficient, α. This material was aligned using rubbed PVA on the glass surfaces. The Williams domains can be observed directly from the scattered light and in detail through a microscope. On the other hand the banded instability were observed using cross polars as alternate bands correspond to 2π and -2π twists separated by thin dark lines of zero twist.

![Fig. 1. Stability curves for a 30 μm cell. The upper curve shows the lower limit of stability of the band made as a function of frequency. The lower curve is for patches.](image-url)

The results shown in figure 1 are the critical shear amplitudes plotted as a function of frequency for a 30 μm layer. The lower curve corresponds to the appearance of patches of Williams domains and the upper for the banded structure. The estimate of the critical amplitudes are obtained by fixing the frequency and increasing the amplitude of the vibration in small steps until the instability is observed. Each instability has a rapid growth rate and so determination of a critical point is relatively straightforward. In addition, there is no evidence of any hysteresis in the transition and so the critical value can be determined by the appearance or disappearance of the instability.

The strong dependence of the critical amplitude on frequency for both instabilities is obvious, and their representative set is typical for all observations in the range 10-50 μm. For layers thinner than 20 μm the bands appear before the patches with increase in amplitude and vice versa for layers thicker than 25 μm. The exchange of priority between the two types of instability is an area of current investigation.
As observed by Clark et al the bands fill the whole cell and have the form of cm-wide stripes separated by thin dark lines and are orientated along the direction of the shear. The patches are generally directed in the same way but they never fill the whole domain, i.e. the patches are separated by regions which have no apparent structure.

The number of bands or patches is dependent on the applied frequency and thickness of the cell. Each of the states are repeatedly formed over a range of frequencies with quasistatic increase of amplitude. The exchange of priority between neighbouring states involves hysteresis and multiplicity of states, i.e. two different patch or band structures at the same supercritical parameter values. An example of this exchange process is sketched in figure 2 for the change over between two and three patches.

Fig. 2. Schematic of the exchange between two and three patches for a 50 μm cell (a) 15-45 Hz (2 patches) (b) 45-75 Hz (mixed mode) (c) 75-80 Hz (mixed connected mode) (d) 80-100 Hz (3 patches).

Over the lower frequency range, two patches are formed while at higher frequencies three are produced with slow increase in amplitude. The intervening range is covered by the interesting mixed mode behaviour. Starting from the low frequency end, four small patches are formed and one of the pair on a diagonal link across, swing through 45° and the other two small patches grow to form a three patch configuration. The process is symmetrical so that the growth along either diagonal is found.

Finally, we show in figure 3 photocopies of photographs of the two types of mechanical Williams domains formed in the patches. The ones shown in figure 3(a) are aligned at right angles to the shear and have a wavelength of approximately two layer thickness depending weakly on frequency. The second type shown in figure 3(b) are aligned in the direction of the shear and have a length scale approximately three times shorter than those shown in 3(a). They are formed if the layer is left in a supercritical state for periods of minutes and seem to arise from an interaction with the weak mean flow field. Once areas of the cell have been reorientated like this then the orientation of the Williams domains becomes the preferred structure for periods of several hours.

Acknowledgements - The main part of this research is supported through the SERC 'Nonlinear Initiative'. GEC Research at Hirst Research Centre provided helpful additional support. Finally, Mr. Nick Porter gave valuable assistance with some of the experimental work.

REFERENCES


DEFECT MEDIATED TRANSITIONS BETWEEN CONVECTIVE STATES IN A NEMATIC LIQUID CRYSTAL: EXPERIMENT AND SIMULATION

JOETS ALAIN
Laboratoire de Physique des Solides
Université de Paris–Sud
91405 Orsay Cedex, France

Abstract: Hydrodynamical systems under constraint constitute rich examples of pattern-forming systems governed by nonlinear-differential equations. The basic solutions may be either ordered states, or inhomogeneous states including (structural) defects. For instance, the convection in a nematic liquid crystal may exhibit a series of structures leading to chaos. We study here a direct transition between two structures of different symmetry and we show that the defects play an important role in the transformation. The complex evolution of the physical system is satisfactorily reproduced in numerical simulations using an appropriate nonlinear model of a so-called “amplitude equation”.

I. INTRODUCTION

A nematic liquid crystal subjected to an AC electric field constitutes an example of a pattern-forming system that can exhibit ordered states as well as “complex states”. As the amplitude of the applied field is continuously varied, the system undergoes a series of bifurcations (transitions) to ordered convective structures of decreasing symmetry: the Normal Rolls (NR), the Oblique Rolls (OR), Varicose, Bimodal, and finally the full chaotic state [1]. Each structure is characterized by a global symmetry. As is the case for every ordered structure, typical defects of the ordering exist and they appear as local states whereby the symmetry is broken. The topology of the defects is related to the broken symmetry. For instance, it is observed that the Normals Rolls have one type of defect: the edge dislocation. It consists of an extra pair of rolls added to the structure at some point (the core).

II. TRANSITION FROM THE NORMAL ROLLS TO THE OBLIQUE ROLLS INDUCED BY THE DEFECTS

We have recently found a direct transition from the first structure, the Normal Rolls, to the second one, the Oblique Rolls, when two control parameters are simultaneously varied [2]. The evolution of the structure is first characterized by a growing undulation of the rolls along their axis (modulational instability). Then, dislocation pairs nucleate in the bending zones of the rolls, leading to the formation of small Oblique Rolls domains. As a consequence of the glide motion of the defects (motion perpendicular to the roll axis) the size of the OR domains grows and very large OR domains are finally obtained (OR structure). In this process, the role of the defects is clearly related to their complex dynamics, which appears as a local wavelength selection mechanism (local tilt of the rolls) and constitutes an efficient mechanism leading to a fast transition to the OR structure.

III. NUMERICAL SIMULATION OF THE TRANSITION FROM NORMAL ROLLS TO THE OBLIQUE ROLLS

This complete transition is actually untractable as the basis of the “microscopic equations”. However, there exists an alternative approach based on the so-called amplitude equations [3,4,5]. This nonlinear model describes the evolution of the 2-D envelope of the rolls, supposed to vary slowly on a large domain with respect to the wavelength of the structure. A dislocation corresponds then to an isolated zero of the (complex) amplitude. For some values of the parameters in the model, and of the initial state, we are able to simulate the observed NR → OR transition. In particular, we reproduce the initial modulational instability, as well as the nucleation of the defects, which by their glide motion are responsible of the formation of the OR domains.

IV. CONCLUSION

The convection in a nematic liquid crystal provides a very pertinent model for the study of extended nonlinear systems. The defects are not merely a local loss of order inside a well-defined structure. They may play a very important role in the structural transition between two states of different symmetry. We show also that the “amplitude equations” are an efficient nonlinear simple model to simulate, at least qualitatively, such a complex behavior.

1 This work was supported by the Direction des Recherches et Etudes Techniques under contract DRET/ERS/901616. The numerical simulations on a CRAY-2 computer were made possible by the Scientific Committee of the CCVR at Palaiseau (France).

Abstract. The classical theory of FRANK explains fairly well point defects that occur in the equilibrium configurations of nematic liquid crystals, but it fails to describe both disclinations and surface defects. ERICKSEN recently proposed a model that aims to accommodate all defects in a unified theory. It is shown here by example how disclinations and surface defects fit into ERICKSEN's model. In both cases a defect arises when a material

Liquid crystals are fluids that exhibit a preferred orientation of the optical axis which varies in space: such a direction is the optical axis of the material. The orientation of the optical axis is customarily described by a unit vector field \( n \). Here I shall conform to such a tradition, but I warn the reader that employing \( n \) to describe the orientation of the optical axis might be inappropriate, especially when one wishes to model general defects, that is, regions in space of any dimension, where \( n \) suffers discontinuities. The orientation of the optical axis, that one is to represent \( n \), is unaffected by a reflection that changes \( n \) into \(-n\) everywhere. Thus, if \( n \) is reflected, say, across a surface, a fictitious defect arises since the optical axis is just the same on the two sides of the surface, while \( n \) suffers a jump.

A way to fix this up might be to describe the optical axis through the tensor field defined by

\[
N := n \otimes n
\]

In general the fields \( n \) and \( N \) are not equivalent in the whole region where they are defined. We denote by \( E \) the three-dimensional Euclidean space and by \( V \) its translation space. Let \( B \) be the region of \( E \) occupied by the liquid and let \( n \) be a field of \( B \) into \( S^2 \), the unit sphere of \( V \). By \( (1) \) one readily associates to \( n \) a field \( N \) of \( B \) into the manifold \( N \) of all symmetric, rank-one tensors whose trace is 1. On the contrary, if a field \( N \) of \( B \) into \( N \) is given, in general one cannot retrace any field \( n \) of \( B \) into \( S^2 \) which is related to \( N \) through \( (1) \), without introducing fictitious defects.

For the problems I review here the fields \( n \) and \( N \) are globally equivalent. Thus, I will stick with \( n \), though I hold that a description of the kinematics of liquid crystals in the manifold \( N \)-wants.

The degree of orientation of a nematic liquid crystal is a scalar \( s \) that ranges in the interval \([-1,2]\). It is expained in Section 2 of [1] how the formal definition of \( s \) can be derived from the statistical distribution of molecular orientations. Qualitatively, \( s \) measures at a macroscopic scale the degree of microscopic order, i.e., it may vary in space. The end-points of the real interval \([-1,2]\) represent two ideal situations. When \( s = -1 \), all the molecules, which resemble rods, are orthogonal to \( n \), but do not lie in any preferred direction, when \( s = 1 \) all the molecules are parallel to each other. When \( s \) vanishes the orientation of the molecules is completely disordered. At a macroscopic scale, this means that the liquid crystal has become an isotropic fluid and \( n \) makes no sense at all.

When, besides \( n \), also the degree of orientation \( s \) comes on the scene, the free energy per unit volume \( \sigma \) is allowed to depend on both \( s \) and its gradient, besides \( n \) and its gradient. ERICKSEN has discussed in [1] a general formula for \( \sigma \); here we employ a special case of that formula:

\[
(\sigma) = \kappa (|\nabla s|^2 + s^2 |\nabla n|^2 + \psi(s))
\]

both \( \kappa \) and \( k \) are positive constants, \( \psi \) a function which describes a double-well potential having two local minimums at \( s = 0 \) and the absolute maximum at \( s = s_0 > 0 \). The main qualitative features of \( \psi \) are described, for example, in Section 2 of [2].

Though there is enough evidence that \( \psi \) often prevails on the other terms of (2) (cf. [3], for example), I shall omit it in (2) and let its role be played instead by a condition that sets \( s = s_0 \) wherever in \( \partial B \) \( n \) is prescribed.

When \( s \) is constant (2) becomes the one-constant approximation to FRANK's classical energy that has been widely studied. FRANK's theory has been successful in solving a number of problems, but defects other than point defects escape its scope (see [4] and [5]).

On the other hand, line defects, also called disclinations, are often observed in ordinary liquid crystals, while surface defects occasionally occur in polymeric liquid crystals. The want for a unified theory of defects prompted ERICKSEN to amend FRANK's theory. In the new theory the spatial changes in the degree of orientation prevent the free energy from being highly concentrated about defects; these are indeed to be identified with the regions where \( s \) vanishes: there the liquid crystal becomes isotropic and the optical axis has no meaning whatsoever. Thus, the new theory interprets defects as localized transitions of the liquid crystal to its isotropic
phase, although no change in the temperature is involved.

The degree of orientation \( s \) and the optical axis \( n \) are delivered in \( B \) by the mappings

\[
s: B \to [-1/2, 1], \quad n: \partial B(s) \to S^2,
\]

where \( S(s) := \{ p \in B : s(p) = 0 \} \) is called singular set.

The points of the singular set are the only sites where defects of \( n \) actually occur (cf. [6]). Thus, if \( S(s) \neq \emptyset \), we say that \( n \) is singular, otherwise we say that it is regular.

The total free energy is the functional

\[
F[s, n] := \int_B \left( k_s |\nabla s|^2 + s^2 |\nabla n|^2 \right).
\]

The existence of minimizers for \( F \) and the degree of their regularity has been established by AMBROSIO (see [7] and [9]) and LIN (see [6] and [8]). Building upon their work, we consider now two variational problems for \( F \) whose solutions describe a disclination and a surface defect, respectively.

Let \( B \) be a circular cylinder and let \( D \) be the lateral boundary of \( B \). We prescribe both \( n \) and \( s \) on \( D \) as follows

\[
n = e_r, \quad s = s_0 \quad \text{on } D.
\]

These boundary conditions are axisymmetric, and so one would expect that the minimizers of \( F \) subject to (3) are axisymmetric too. Such a conclusion is in general false, as is shown in Section 5 of [10], but here one can prove that the minimizers of \( F \) within a quite broad class (which also includes non-axisymmetric fields) are indeed axisymmetric (see [11]). This variational problem was solved in [12].

There we reached the following conclusion:

If \( k \leq 1 \) then \( F \) subject to (3) attains its minimum when \( n \) is radial everywhere in \( B \) and \( s \) vanishes along the axis of \( B \), where there is a disclination. If \( k > 1 \) then the field \( n \) that minimizes \( F \) subject to (3) is regular and closely resembles the celebrated solution of CLADIS & KLEMAN [13], which is fluted along the axis of the cylinder.

Let now \( B \) be the region of \( E \) between two square parallel plates. We call \( D^+ \) and \( D^- \), respectively, the plates that bound \( B \); we set \( D = D^+ \cup D^- \). We prescribe \( n \) and \( s \) on \( D \) thus

\[
n = e_1 \quad \text{on } D^+ , \quad n = \cos \alpha_0 e_1 + \sin \alpha_0 e_2 \quad \text{on } D^- , \quad s = s_0 \quad \text{on } D,
\]

where \( \alpha_0 \in ]0, \pi[ \) and \( e_1, e_2 \) are unit vectors directed along two adjacent sides of the plates. The variational problem for \( F \) subject to these boundary conditions was solved in [14]. Here is the conclusion we reached there.

If \( k \leq (\sqrt{\alpha} / \epsilon)^2 \) then \( F \) attains its minimum when \( n \) is discontinuous at the mid plane between \( D^+ \) and \( D^- \), and constant elsewhere. If \( k > (\sqrt{\alpha} / \epsilon)^2 \) then the field \( n \) that minimizes \( F \) is regular and represents a non-uniform twist. Thus, for \( k \) sufficiently small a surface defect arises between two adjacent domains.

The effect of a potential like \( \psi \) on the occurrence of such a defect has been studied in [15] within a special class of minimizers: the qualitative features of the phenomenon seem to be unaffected by \( \psi \), but the critical value of \( k \) decreases.

References


Abstract - The 4X4 matrix method of computing reflection and transmission by layered structures can be very fast if a few simple mathematical and programming procedures are followed to enhance efficiency. Failure to utilize these procedures has led to a misconception that the 4X4 matrix method is extremely time consuming. Structures with n-1 regular periodic variations parallel to the surface can be treated approximately using a 4nX4n extension of the method, to which most of these procedures for enhancing efficiency also apply.

INTRODUCTION

The 4X4 matrix method is a generalization, for anisotropic media, of the Abell's double-2X2 matrix method for computing reflectance and transmittance of plane, isotropic layered structures. Since the 4X4 method was first used to compute reflection and transmission in twisted liquid crystal structures there have been continued attempts to find faster ways to solve the problem. However, the fact that many groups use the 4X4 method regularly to design twist- and super-twist cells with phase-retarding plates, color filters and polarizers all in the structure, attests to its versatility if it is efficiently programmed.

Closed-form solutions for isotropic media are and for uniaxial and biaxial materials in certain special orientations have been known for many years. It was also mentioned in several early publications that the transfer matrix for a uniform slab of any thickness could be obtained through solving for the eigenvalues of the 4X4 differential transfer matrix, to obtain the propagation vectors of the four optic modes in the medium. The secular equation for these modes was given in compact form by Teitel and Henvis, Eq.16. Wohler et al. recently published the solution to the secular equation in closed form, and the resulting elements for the 4X4 transfer matrix, for a uniform layer of uniaxial material of any thickness and any orientation. The solutions has recently been termed the "fast" method.

Most nematic twist cells can be reasonably well approximated by using rather few, thick sub-layers of material. Most polarizers are also uniaxial, as are most retarding films. The "fast" method is advantageous in these regions.

If the liquid crystal in the cell is divided into very many short segments for high precision the "slow" method is faster than the "fast" because eigenvectors are not computed. If ferroelectric smectic displays and nematic displays with high field gradients are of interest, then biaxiality may be significant. The "fast" method is then tedious and slower.

Two misconceptions discourage people from trying the "slow" method, despite its simplicity and possible speed. The first is the supposition that it is necessary to divide the structure into segments of thickness \( T \) that are so thin that the square of the phase change of the wave across each segment is negligible: less than the wavelength in the medium, \( \lambda \), divided by \( 2 \pi \times 10^4 \). The second is that one must therefore multiply 4X4 matrices 62,800 times just to get through a thickness of one wavelength within the medium.

The first misconception at the outset of my work with J.Schaeffer. However, I used the "method of repeated squaring" to greatly reduce the number of matrix multiplications. This method is still the answer to the second objection. Without any other improvements the number of matrix squarings required to get through \( T \) would be about \( \log_2(T/62,800) \), and finally store the \( n \) sums. For the case of 8X8 matrix optics, the number of multiplications, which take most of the time, is 2048. Actual measurement confirms that matrix multiplication is the most time-consuming part of the fastest 4X4 programs that were made. Namely, made the "slow" method, but when \( n=1 \), (4X4 matrix optics), it is usually only somewhat longer than the rest of the computation.

Many 4X4 multiplications are to be done even if individual transfer matrices are to be found by the "fast" method. Hence it is important, in any case, to write an efficient matrix multiplication program. With most compilers a great saving in time is achieved by writing out the 4X4 matrix product elements explicitly; avoiding loops and computed indexes.

B: Repeated Squaring Technique

Let \( T \) designate the thickest layer for which the transfer matrix can be computed in "one step" without making significant approximation errors. This thickness depends on the method used, as mentioned before. The next problem is to obtain the transfer matrix \( P(T) \) over a slab of thickness \( T > T \) using the relation \( P(T) = P(T)P(H) \) adroitly, where \( H < T \). The fastest method is to set \( P(T) = \sum (P(H)) \), where \( n \) is the smallest integer larger than \( \log_2(T/T_m) \). Of course, if \( T > T_m \) then \( n = 0 \), and no squaring is necessary.

C: Matrix Differential Equation and Exponential Series Expansion for a Thin Layer

The 4X4 matrix method starts with a matrix differential equation, equivalent to Maxwell's equations, where \( \Delta \) is a 4X4 differential matrix whose elements depend only on the direction of propagation of the incident beam and the possibly complex optical dielectric tensor, \( \varepsilon \), at that point in the multilayer. The principal values of \( \varepsilon \) are the squares of the principal refractive indexes, \( n_1^2 + k_1^2 \). The four-element "vector" contains the four electric and magnetic field components parallel to the surface of the layered structure, or normal to the "z" axis.

Let \( S \) be the sine of the angle of incidence of light from vacuum, and assume that \( S \) is symmetric.
Then the differential matrix is $\Delta = i^d/e_{zz} =$

$$
\begin{bmatrix}
-s_{xx} & e_{zz} - s^2 & -s_{xy} & 0 \\
-e_{xzxz} & -s_{xx} & e_{yy} & 0 \\
0 & 0 & 0 & 1 \\
D_{23} & D_{11} & (e_{yy}-s^2)e_{zz} & e_{xz}e_{yz}
\end{bmatrix}
$$

and $\Psi = \begin{bmatrix} E_x \\ E_y \\ E_z \\ -H_y \end{bmatrix}$.

We compute the $P$-matrix from the last surface to the first. This avoids the necessity of doing a matrix inversion at the end, which saves significant computer time and programming effort. If $z$ increases in going from the entrance to the exit side of the layered structure then the solution to the matrix differential equation in a region of invariant optical dielectric tensor of thickness $H < T_m$ is $^{1,4,5}$

$$
\Psi(2-H) = P(-H)\Psi(2) = [1 - h\Delta + (h\Delta)^2/2! - (h\Delta)^3/3! + (h\Delta)^4/4! - \cdots]\Psi(2),
$$

where $h = 2\pi H/\lambda$. $P(-H)$ is the transfer matrix relating the field $\Psi$ at the side of the layer where the beam enters to that at the exit side, and $h^2\Delta_{11}\Delta_{13}\cdots\Delta_{33}$ defines the $ij$ element of $(h\Delta)^n$. Accuracy of one part in $10^8$ may be achieved if no element of $h\Delta$ has magnitude larger than about unity, and if the series runs to $(h\Delta)^{11}/11!$.

To find an appropriate value of $h$ for a uniform layer of thickness $T$, first find the square of the matrix, $\Delta^2 = \Delta_1 \Delta_3$. (This squared matrix is used again in the two series to follow, so it is not wasted.) Find the absolute square of $\Delta_{11}\Delta_{13}$ and of $\Delta_{31}\Delta_{33}$ and select the larger. This number is the magnitude of the largest element likely to appear in $\Delta^2$. Call its fourth root $\Delta^*$. The series will converge rapidly if $h\Delta^* < 1$. Count the number of times, $n$, that the layer thickness $T$ must be halved before $H = T/2^n < 1/(2\Delta^*)$. Then $n$ is the number of times $P(-H)$ must be squared to get $P(T)$, and $h = 2\pi H/\lambda$. To achieve the eleventh order expansion with six matrix multiplications, write

$$
P(-H) = 1 + (h\Delta)^2/2! + (h\Delta)^4/4! + (h\Delta)^6/6! + (h\Delta)^8/8! + (h\Delta)^{10}/10! - (h\Delta)^{12}/12! + (h\Delta)^{14}/14! + (h\Delta)^{16}/16! + (h\Delta)^{18}/18! + (h\Delta)^{20}/20! + (h\Delta)^{22}/22! + (h\Delta)^{24}/24! + (h\Delta)^{26}/26! + \cdots.
$$

If $T < 1/(2\Delta^*) < T_m$, then the preceding two series may be truncated. Accuracy of one part in $10^8$ is maintained when the tenth power terms are neglected if $\log(2T/T_m) < -0.9$; the eighth power if $< -2.0$; and the sixth if $< -7.5$. If the medium does not absorb light, then $\Delta$ is a real tensor or scalar and $\Delta$ is then purely imaginary. Hence $\Delta^2$ is a matrix of real numbers. Additional speed may then be obtained in such media by writing a separate program to multiply real 4X4 matrices. Such a program is nearly four times faster than one for complex numbers. However, the special program is useful only for expanding the power series, not for subsequent repeated squaring.

II: Abelès’ Method for Isotropic Regions

For isotropic media, $e_{xx} = e_{yy} = e_{zz} = e(n+1)^2$ and $e_{N7} = e_{y7} = e_{xz} = 0$. Hence the local 4X4 matrices, $\Delta$ and $P(T)$, have zeros in the off-diagonal 2X2 quadrants. The 4X4 matrix is then composed of Abelès’ two 2X2 matrices along the diagonal:

$$
\Delta = \begin{bmatrix} 0 & 1 -(s^2/\delta) & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & e^{-s^2/\delta} & 0 \\ 0 & 0 & 0 & e^{-s^2/\delta} \end{bmatrix}
$$

The square, $h^2\Delta_{ij}\Delta_{kj}$, is just a constant, (complex if the medium absorbs light), multiplied by the unit 4X4 matrix. That constant is $h^2(\Delta_{12} \Delta_{21}) = h^2(\Delta_{34} \Delta_{43}) = h^2(e - s^2)$. The two series reduce to power series in this (possibly complex) number. The first series generates the cosine and the second (except for a factor) the sine of the number. These are the Abelès closed-form solutions. No matrix multiplication is required to obtain the 4X4 transfer matrix in this case, but the magnitude of the constant should be less than unity, as in the more general case of the 4X4 matrix.

One may prefer to use pre-programmed functions for the complex sines and cosines, which may or may not take account of the slow convergence problem for large arguments.

There is nothing comparable to Abelès’ method for 8X8 or higher-order matrix optics. The possibility of a step between the incident beam and the direction of the lateral periodic structure leaves open the possibility of non-zero elements in any region of the matrix.

CONCLUDING REMARKS

We find that each computation of reflectance and transmittance by a supertwist nematic liquid crystal cell takes 1.53 seconds using the "fast" meth. on a math co-processor, when the following parameters are used: The cell has two 1-mm glass cover plates, two absorbing uniaxial polarizers, two absorbing InSb oxide conductive layers, a uniaxial compensating fractional-wave plate, and the non-absorbing liquid crystal is subdivided into 36 layers. The "fast" method would enhance the speed somewhat in going through the polarizers and retarding plate, but less in going through the much-subdivided liquid crystal.

If one had a computer with an array processor, computing the transfer matrix for a uniform slab might be about as fast as computing a sine, cosine or exponential on a single-processor computer. In that case the "slow" method might be somewhat faster than the "fast" method of 4X4 matrix optics, especially for biaxial materials, since the eigenvector problems would be avoided at each different layer. However, an array processor could probably be used to better advantage in liquid crystal cell simulation to run several different wavelengths or directions of incidence simultaneously. For 6X8 and higher order matrix diffraction problems the potential for faster matrix multiplication with an array processor might enhance the speed considerably.

REFERENCES

1) D.O. Smith, Optica Acta 12,13(1965)
4) Florin Abléès, Ann. de Physique 5,596(1950)


Abstract

Numerical calculations of director configurations and electrooptical characteristics in symmetrical and nonsymmetrical nematic layers are presented, when weak anchoring in the tilt and twist angle of the director is assumed. In cells with bistabilities we investigate the influence of the anchoring parameters and device parameters on the width of the hysteresis. Using the 4x4 matrix formalism of Blan
erman, we demonstrate the influence of the weak anchoring on the transmission vs. voltage characteristic and CIE color coordinates.

Our main emphasis was to implement an efficient code which works properly and fast over a broad range of material and device parameters.

1 Introduction

The director configuration in twisted nematic layers like TN[1], OMI[2], or SBE[3]-cells is determined by the following three major features: first, the elastic forces in the liquid crystal, described by the well known Frank-Oseen-Zocher free energy density. Second, the influence of an external applied voltage modeled through an electric energy term of the form \( \frac{1}{2} D^2 E \), where \( D \) is the displacement vector and \( E \) the internal electric field vector, and third, the anchoring of the director at the substrate boundaries of the layer. Recently, some experimental studies of the anchoring have been published[4, 5, 6], showing that typical values of the anchoring energy are in the range \( 10^{-6} N/m \) to \( 10^{-5} N/m \) for homoeotropically anchored nematics and \( 10^{-7} N/m \) to \( 10^{-5} N/m \) for planarly oriented nematics. In the last case, the twist anchoring energy is one order of magnitude smaller than the tilt energy.

In the following, the two kinds of anchoring are combined and studied at the influence of the hysteresis width and on the electrooptical properties in symmetrical as well as in nonsymmetrical cells.

2 Theory

We consider a nematic cell of thickness \( d \) located between the planes \( z = 0 \) and \( z = d \) of a Cartesian coordinate system. The director \( n \) is described by the tilt angle \( \theta \) (measured from the layer normal) and the twist angle \( \varphi \). The dielectric constants are denoted by \( \varepsilon_0 \) and \( \varepsilon_1 \). The elastic constants for splay, twist and bend are denoted by \( k_{11}, k_{22} \) and \( k_{33} \) respectively. The pitch of the material induced through a chiral dopant is named \( p_0 \).

Using the abbreviations

\[
\begin{align*}
\alpha_1 &= k_{33} \cos^2 \theta + k_{11} \sin^2 \theta \\
\alpha_2 &= (k_{33} \cos^2 \theta + k_{22} \sin^2 \theta) \sin \theta \\
\alpha_3 &= \frac{k_{22}}{k_{33}} \sin^2 \theta \\
\alpha_4 &= \varepsilon_0 (\varepsilon_1 + \Delta \varepsilon \cos^2 \theta),
\end{align*}
\]

the free energy density in the bulk can be written as:

\[
\begin{align*}
\Phi_B &= \frac{1}{2} \alpha_1 \phi'^2 + \alpha_2 \phi'^2 - \alpha_3 \phi' - \frac{1}{2} \alpha_4 \phi'^2.
\end{align*}
\]

where the prime indicates differentiation with respect to \( z \) and where \( \phi \) is the electric potential inside the layer. Note that the last term represents the electric contribution \( \frac{1}{2} D^2 E \), when we assume a dielectric material law for uniaxial nematics in the form:

\[
D_1 = (\varepsilon_1 \phi' + \Delta \varepsilon \phi' n_j) n_j,
\]

with \( E_1 = -\phi \).

The weak anchoring in the tilt and twist angle at the top and bottom of the cell is described by a surface free energy of the Rapaport-Papoular type[7]:

\[
\begin{align*}
\Gamma_s^2 &= \frac{1}{2} C_0^2 \sin^2 (\theta(0) - \phi^2) + \frac{1}{2} C_0^2 \sin^2 (\phi(0)) \\
\Gamma_t^2 &= \frac{1}{2} C_0^2 \sin^2 (\theta(d) - \phi^2) + \frac{1}{2} C_0^2 \sin^2 (\phi(d) - \phi_f)
\end{align*}
\]

The factors \( C_0^2 \) resp. \( C_v^2 \) measure the anchoring strength in the tilt and twist angle respectively, \( \phi_f \) (pretwist) describe the preferred tilt angle of the director at the surfaces, \( \phi_f \) (pretwist) is the difference between the preferred orientations in the twist angle at the top and bottom surface. The influence of the surface is restricted to the place of the aligning substrate.

The total free energy per unit area of the cell is now given by:

\[
\Phi = \int_0^d \Phi_B dz + \int_0^d \Phi_F du
\]

3 Numerical Procedure

The first step in the calculation of optical properties consists in the determination of the director configuration. To this end, we transform the Euler-Lagrange equations resulting from the extremalisation of the bulk free energy into a system of Hamilton equations by performing a Legendre transformation with respect to the variables \( \theta, \phi \) and \( \Phi \):

\[
\begin{align*}
\theta' &= \frac{\partial \Pi}{\partial \theta} = T \\
\phi' &= \frac{\partial \Pi}{\partial \phi} = P + a \phi - \Phi \\
\Phi' &= \frac{\partial \Pi}{\partial \Phi} = -U \\
T^2 &= -\frac{\partial^2 \Pi}{\partial \theta^2} = \frac{1}{2} \varepsilon_0 \frac{1}{\varepsilon_1 + \Delta \varepsilon \cos^2 \theta} \frac{\partial^2 \Phi_B}{\partial \phi'^2} \\
P^2 &= -\frac{\partial^2 \Pi}{\partial \phi^2} = \frac{1}{2} \frac{\partial^2 \Phi_B}{\partial \phi'^2} \\
U^2 &= -\frac{\partial^2 \Pi}{\partial \Phi^2} = \frac{1}{2} \frac{\partial^2 \Phi_B}{\partial \phi'^2}
\end{align*}
\]

where \( \Pi \) is the Legendre transform of \( \Phi \):

\[
\Pi(\theta, T, P, U) = \theta T + \phi P + \Phi U - \Phi_B,
\]

and \( b_n \) is the Legendre transform of \( b_n \):
and the corresponding equations at $z = d$.

The equations (6) and (9) represent a nonlinear boundary value problem, which we solve numerically by a multidimensional shooting method using standard library routines.

The second step is to solve the Maxwell equations inside the layer. For this problem we use the $4 \times 4$ formalism of BERREMAN[8, 9].

On an IBM PC/AT with 10 MHz, for instance, the calculation of an electrooptical characteristic needs about ten minutes independent of the existence of bistabilities.

4 Results

4.1 Director Configurations

In fig. 1, we show director configurations for a symmetrical and nonsymmetrical $90^\circ$-TN-cell with $\theta_p = 89^\circ$ and $C_{ij}^P = 10^{-9}$N/m in both cells; $C_{ij}^P$ is $10^{-32}$N/m whereas $C_{ij}^2$ is $10^{-32}$N/m for the nonsymmetrical cell. The parameter for the curves is the applied voltage.

With increasing voltage, the director tries to align parallel to the applied electric field, e.g. the tilt angle decreases, whereas the twist becomes more and more nonlinear.

4.2 Width of the Hysteresis

In highly twisted cells with nonzero pretilt, there is the possibility of bistable director configurations[10], which becomes evident by a hysteresis in the $\theta_m$ vs. voltage curve. In fig. 2, we have plotted the width of the hysteresis $\Delta V$ for a symmetrical cell as a function of $C_{ij}$ and $\phi_p$ for fixed $C_{ij} = 10^{-9}$N/m and $\theta_m = 15^\circ$; $\phi_p$ is chosen in such a way, that it matches $\phi_p$. We get a monotonic increase of $\Delta V$ in $\phi_p$. Further, it can be seen that for fixed $\phi_p$, the width of the hysteresis falls with decreasing $C_{ij}$.

4.3 Transmission vs. Voltage Curves, Color Coordinates

One of the most important features for a twisted nematic layer is its transmission vs. voltage characteristic. In fig. 3 (left), we show these curves for a $90^\circ$-TN-cell for which the surface parameters are the same as in section 4.1, except that we vary $\theta_p$.

It is seen, that an increase in the asymmetry of the cell decreases the optical threshold voltage. The color coordinates of the cell with the applied voltage as parameter are shown in the right of fig. 3. The cell with the greatest asymmetry gives smallest color changes.
Abstract. Within the general field of pattern formation studies, one of the most interesting situations corresponds to the analysis of a transient dynamical evolution to a stable steady pattern. Here we will deal with a particular realization of such a process which connects two homogeneous steady states through a transient inhomogeneous structure. The problem will be here addressed in relation with the Freedericksz transition in nematic liquid crystals. The general nematodynamic equations, incorporating hydrodynamic contributions and internal degrees of noise, will adopt the form of a Time Dependent Ginzburg Landau model. The pattern dynamics is monitored in terms of the temporal evolution of the structure factor for the orientational fluctuations.

I. INTRODUCTION

The magnetically induced Freedericksz transition occurs in a nematic slab when the director, describing the state of orientation of the nematic molecules inside the sample, reorients following an applied magnetic field larger than a critical one $H_c$. The standard description corresponds to the appearance of distortions in the orientation with respect to the original one, the degree of distortion being homogeneous in each plane of the sample, and of maximum intensity in the mid plane far from the plates limiting the nematic material [1].

This simple picture of the Freedericksz instability is however too simplified to account for some experimental observed facts. Under appropriate conditions, transient spatial structures are found corresponding to modes of inhomogeneous distortions in the planes of the sample. A wide experimental evidence of this phenomenon, [2,3], supplemented with detailed theoretical analysis, [4,5], have been accumulating during these past years. The suggested explanation involves a dynamical coupling between the director field and the hydrodynamic motion associated with the reorientation. Such a coupling gives rise, during the transient process, to spatial domains with a well-defined periodicity.

The characteristic wavenumber of these transient patterns has been commonly described in terms of a most unstable mode. A linear analysis of the nematodynamic equations around the initial undistorted configuration identifies the mode of fastest growth. It is assumed that this mode dominates the transient dynamics. However, this approach although useful in understanding the main physical ingredients involved in the phenomenon, is far less valid if one is interested in the dynamics of the pattern formation process itself. With this specific aim we have recently proposed a model, [4], which permits us to identify the basic time scales governing the reorientation of the nematic sample.

The analysis is based on the evolution equation for the time-dependent structure factor which describes the orientational distortions of the director, once thermal fluctuations and hydrodynamic effects have been taken into consideration. A central role is played by an effective viscosity which accounts for a tradeoff of rotational for shear viscosities leading to a compromise at some intermediate nonzero wavenumber for which the increase in elastic energy contribution is favorably balanced by a higher energy dissipation rate. In what follows we will describe this behavior for the simplest realizations of the magnetically induced Freedericksz transition.

II. PATTERN FORMATION EQUATIONS

The simplest geometry we can envisage corresponds to a twist geometry. In this situation the sample is contained between two plates perpendicular to the z axis. The director is initially aligned along the x axis, and the magnetic field is applied along the y axis. The transient behavior we will describe corresponds to the switch at $t=0$ from an initial value $H_o < H_c$ to a final one $H_o > H_c$. Under these conditions it has been experimentally shown that in addition to the usual twist deformations along the z axis, the system may transiently develop a more complicated structure involving bend modes along the x direction (Fig.1).


The appropriate scheme of nematodynamic equations is here used assuming for simplicity that macroscopic flow exists only along the y direction, and that homogeneity extends on the direction of the applied magnetic field. For typical on-plane reorientations we introduce the dinamical variable $z$ as

\[
\begin{align*}
\eta_1(x,z) &= \cos(\phi(x,z)), \\
\eta_2(x,z) &= \sin(\phi(x,z)), \\
n_z &= 0.
\end{align*}
\]

A minimal coupling approximation is then invoked to obtain a pair of closed equations for $\phi$ and $\eta_z$

\[
d_t \phi = \frac{1}{r} \frac{1}{2p} \left[ \frac{1}{2} (1 + \lambda) \phi_x \right] + \frac{1}{2p} \left[ \eta_z \right] - \frac{1}{2p} \left[ \eta_z \right]
\]

\[
\frac{d}{dt} \frac{\partial \phi}{\partial r} = \frac{1}{p} \left[ \frac{1}{2} (1 + \lambda) \phi_x \right] + \frac{1}{p} \left[ \eta_z \right] - \frac{1}{p} \left[ \eta_z \right]
\]
The Gaussian random forces appearing in the above Langevin-type equations satisfy fluctuation-dissipation relations in terms of pure rotational and shear viscosities, respectively $\gamma_1$ and $\nu_{23}$. L and $\rho$ stand respectively for a linear transversal dimension and the mass density of the sample. A series of technical manipulations are then strictly convenient. First, one introduces the hypothesis of negligible inertia which enables us to obtain a closed equation for the deformation angle. This equation is more easily handled in a Fourier representation appropriate to the strong boundary conditions prescribed for the nematic at the limiting plates, $z = \pm d/2$:

$$\delta(x) \approx \sum_{m} a_{m} e^{i m x}$$

$$\phi(x) \approx \sum_{m} b_{m} e^{i m x}$$

$$\Omega_{\nu}(x) \approx \sum_{m} \frac{2 \nu_{23} b_{m}}{d} e^{i m x}$$

$$\Omega_{\nu}(x) \approx \sum_{m} \frac{2 \nu_{23} b_{m}}{d} e^{i m x}$$

The resulting equation for the amplitude of the reorientational mode is given by

$$\frac{\partial}{\partial t} \delta_{m}(t) = \left( \frac{1}{\eta_1} \right) \left( \chi_{2} H^2 - K_{23} (2m + 1) \gamma_{1}^{2} \right) \delta_{m}(t)$$

$$+ \frac{2 \nu_{23} b_{m}}{d} e^{i m x}$$

$$Q = \frac{2 \nu_{23} b_{m}}{d}$$

The important point to be noticed is that the temporal evolution of the reorientational process is no longer dictated by the pure rotational viscosity $\gamma_1$ but is governed by an effective wavenumber dependent viscosity $\gamma_{1}(q)$. Thus, the coupling of the director and velocity fields results in a reduction of the viscosity for all modes with $q_{x} \neq 0$. This permits modes of bend deformation along the $x$ direction to grow faster than the homogeneous one, giving rise to pattern structuration. Actually, for $H > H_{c} = (K_{23}2H_{c}^{2}/\chi_{23}^{2})$, $K_{23}$ being the twist elastic constant and the anisotropic diamagnetic parameter, twist modes of wavenumber $m$ become unstable. However due to the dependence of $\gamma_1$ on $q_{x}$, bend modes with $q_{x} = 0$ may lead the response of the system provided that

$$h^{2}(m) > 1 + \frac{K_{23}}{K_{23}} \eta_{1}$$

$$h^{2}(m) = H^2/(2m + 1)H_{c}^1$$

$$\alpha_{2} = -\frac{1}{2} \gamma_{1}(1 + \lambda)$$

$$\eta_{2} = \nu_{23}$$

$$\nu_{23} = \nu_{3} + \frac{1}{2} \gamma_{1}(1 + \lambda)^2$$

where $K_{23}$ is the bend elastic constant and the remaining parameters in (5) are convenient and standard redefinitions of nematic viscosities. Thus, one predicts the appearance of a periodic pattern for magnetic fields satisfying the above condition. This occurs for fields not much larger than the critical one $h^2 = 1$, although there still exists a range of magnetic intensities for which the homogeneous response dominates.

The early dynamical stages of the transition can be easily followed by converting Eq.(4) into an equation for the structure factor. Using standard methods one obtains

$$\frac{d}{dt} C_{m}(t) = \frac{2}{\gamma_{1}} \left[ \chi_{2} H^2 - K_{23} \left( \frac{(2m + 1) \nu_{23}}{d} \right) - K_{23} \eta_{1} \right]$$

$$\times C_{m}(t) + \frac{2j_{0} T}{\gamma_{1}}$$

$$Q = \frac{2j_{0} T}{\gamma_{1}}$$

A convenient way of monitoring the dynamical emergence of the pattern consists in analyzing the time evolution of the mode $Q_{\text{max}}$ corresponding to the maximum of the structure factor for the most unstable twist deformation mode ($m=0$). This is depicted in Fig 2

**FIG.2 Maximum of the structure factor vs. time (taken adimensionalized in units of $t_0 = \gamma_{1}/(\chi_{23} H_{c}^2)$). Parameter values correspond to those of MBBA (Ref.4).**

Different and well-resolved time scales can be distinguished in this figure. A first time scale is associated to the sharp increase of $Q_{\text{max}}$ when the system takes off from the initial conditions. This time should be interpreted as the characteristic time of appearance of the periodic pattern. A second time scale for the slow growth of $Q_{\text{max}}$ is reasonably associated with the development of the inhomogeneous structure. Late stage dynamical scales accounting for the disappearance of such transient patterns can not be described within the limits of this approach and instead, a proper analysis of the mobility and recombination of defect walls is more adequate [6].

Sometimes, the transient nature of the predicted periodicity is already apparent within the linear approximation here used. This is for example the case for slightly supercritical conditions in the splay geometry [7], as shown in Fig. 3. This situation is relevant to recent experiments for the electrically induced Freedericksz instability [8].

**REFERENCES**

We present a numerical computation of director configurations, switching behaviour and optical performance of a nematic liquid crystal (LC) layer in two dimensional inhomogeneous (2D) electrical fields. Coupling between director deformation and electrical field is fully included. The computations are applied to LC spatial light modulators (SLM). For the first time the analogy of the modulation transfer function of SLM is specified. Its dependence upon device and LC material parameters is studied.

I. INTRODUCTION

Usually, a nematic LC display is treated as a stratified medium. The corresponding one dimensional theory is well understood /1/. The optical performance of an LCD is computed in two steps: First, the distribution of the optical axis in the LC layer (director pattern) is determined from continuum theory /3/. Second, the optical properties of the display are computed, commonly with Berreman's 4x4 matrix formalism /2/ for stratified media /1,5/.

In practice the optical response of the LC often depends on a spatially non-uniform electric field, resulting e. g. from pixel structures on matrix displays or intensity distributions on a photoconductor in LC SLMs /6/. With the ever decreasing display structures, director patterns in spatially inhomogeneous fields, limits of spatial and temporal resolution of LC layers and their optical performance become important.

II. CALCULATION OF DIRECTOR PATTERNS

We consider a nematic layer of thickness d between the planes z=0 and z=d of a Cartesian coordinate system under the influence of an electrostatic potential, with boundary conditions for the director tilt angle θ and the potential V:

\[ \theta(x,z=0) = \theta_0 \quad \theta(x,z=d) = \theta_0 \]  
\[ V(x,z=0) = V_0 \quad V(x,z=d) = V_1(x) \]

We restrict ourselves to constant twist angle \( \phi \) and two principal cases of director alignment: in the xz-plane of the field inhomogeneity (\( \theta = 0 \)), and perpendicular to it (\( \theta = 90^\circ \)). We assume invariance in y-direction. Our aims are: 1) the determination of the equilibrium director pattern \( \theta(x,z) \) and corresponding electric potential \( V(x,z) \); and 2) the computation of the director response to sudden changes in \( V_1(x) \). The second problem can be treated as follows: Since the time constant of the electric field is much smaller than that of LC dynamics, we assume the electric field to be quasi-static and for a given director pattern \( \theta(x,z,t) \) the corresponding potential \( V(x,z,t) \) is found from the equations of electrostatics of inhomogeneous media:

\[ (c_1 + \alpha c^2) V''_{xx} + 2 \alpha c \theta \theta_x V'_{x} + V_x \theta_x' + V_x \theta' = 0 \]  
\[ + (c_3 + \alpha c^2) V''_{zz} + \alpha (c^2 e - s^2) (V_x \theta_z + V_z \theta_x) = 0 \]

for \( \phi = 0 \) (3a)

\[ c_1 V''_{xx} + (c_1 + \alpha c^2) V''_{zz} + 2 \alpha c \theta \theta_x V'_x = 0 \]  
\[ \text{for } \phi = 90^\circ \]  
(3b)

\[ V_{ij} = \delta^i_j V_{/i j}, \quad \theta_1 = \theta / \delta \theta, \quad \theta_0 = \sin \theta, \quad c_0 = \cos \theta \]

\[ \Delta c = (c_n - c_0), \quad c_n \]  
and \( c_0 \) are the dielectric constants parallel and normal to \( n \), resp. Now we integrate the nematodynamic equations 4a,b for \( \theta(x,z,t) \) /3/ keeping \( V(x,z,t) \) constant during each integration step. Inertia and backflow are neglected.

\[ \gamma_1 \theta = \gamma_{xx} (K_1 e^2 + K_2 e^2) + \gamma_{zz} (K_1 e^2 + K_3 e^2) \]

\[ + (K_3 - K_1) s \theta \theta_x \theta_z + (K_3 - K_1) (e^2 - s^2) \theta_x \theta_z \]

\[ + c_0 \Delta c (s \theta \theta_x (V_x V_x + V_x V_x) + (e^2 - s^2) \theta_x \theta_z) \]

for \( \phi = 0 \). (4a)

\[ \gamma_1 \theta = \gamma_{xx} (K_1 e^2 + K_3 e^2) + \gamma_{zz} (K_1 e^2 + K_3 e^2) \]

\[ + (K_3 - K_1) s \theta \theta_x \theta_z + c_0 \Delta c \theta_x \theta_z \]

for \( \phi = 90^\circ \). (4b)

The \( K_i \) are LC elastic constants, \( \gamma_1 \) is the rotational viscosity. We repeat this procedure to compute the complete director response using a finite difference method /4/. The resulting set of ordinary equations is solved by a relaxation method. An implicit time integration scheme is applied.

This procedure is applicable as well to the computation of the equilibrium director pattern. However, we prefer a faster approach. Eqs. (3,4) without the dissipative term \( \gamma_i \theta \) are coupled self-consistently. They are solved by a multigrid relaxation method /4/ which speeds up computation remarkably.

III. OPTICAL PERFORMANCE OF A 2D-DEFORMED LIQUID CRYSTAL LAYER

Now we calculate the optical performance of the untwisted nematic layer with a given director tilt profile \( \theta(x,z) \). For normal light incidence and a director tilt profile varying in x-direction slowly compared to the wavelength \( \lambda \) of the light we can use the geometrical optics approximation (GOA) /7/ and describe the light propagation inside the LC layer as locally one-dimensional. The transmitted intensity is

\[ T(x) = \sin^2 \left( \frac{\pi c_0 \theta(x) \cdot n_0}{\lambda} \right) \]

with \( n_0 \), \( n_e \) and \( n_0, \text{eff}(x) \) being the ordinary, extraordinary and effective extraordinary refractive indices of the nematic.
IV. OPTICAL PERFORMANCE AND RESOLUTION OF SLM

We consider a transmissive SLM consisting essentially of LC layer and photoreceptor. Resolution and field fringing of the photoreceptor have been investigated recently by Armitage, Thackara, and Eades, whereas the influence of the LC layer was however neglected. In this paper we study the resolution capability of the LC layer and calculate the optical response to harmonic boundary voltage distributions

\[ V_1(x) = V_0 + \frac{1}{2} V_i(t) \left[ 1 - \cos \frac{2\pi x}{g} \right] \]  

(6)

which are caused by periodic intensity modulated stripe patterns imaged at the photoreceptor. In the following we set the potential at the second electrode \( V_0 = 0 \). For given material and device parameters one has to compute now the electro-optic characteristics between crossed polarizers from the one-dimensional theory. Optimum bias \( V_0 \) and amplitude \( V_i \) are determined by the requirement of maximum contrast and minimum response time. In the computation we used the parameters \( K_1 = 11.3 \text{pN}, K_2 = 8 \text{pN}, K_3 = 12.5 \text{pN}, c_1 = 24.8, c_{i1} = 6.4, \theta_o = 0, n_v = 1.610, n_r = 1.489, \gamma_s = 78.3 \text{mPas}, \lambda = 550 \text{nm}, d = 0.010 \text{mm} \). We compare two possible switching regions: switching between the first two extrema of the transmission vs. voltage curve (\( V_0 = 0.94 \text{V}, V_i = 0.27\text{V} \)), and between the last two extrema (\( V_0 = 1.51 \text{V}, V_i = 0.76\text{V} \)). Figure 1 shows the transmitted intensity as a function of \( x \) for \( \phi = 0 \) and \( \phi = 90^\circ \) and both switching regions calculated according to eq. (5) with an assumed grating constant \( g = 0.020 \text{mm} \).

![Fig. 1. Light transmittance as function of x.](image)

<table>
<thead>
<tr>
<th>Curve</th>
<th>( \phi ) (deg)</th>
<th>Switching Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>First two extrema</td>
</tr>
<tr>
<td>2</td>
<td>90</td>
<td>Last two extrema</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>First two extrema</td>
</tr>
</tbody>
</table>

Ono can see, that imaging is always better for \( \phi = 90^\circ \) compared to \( \phi = 0 \). The \( \phi = 0 \) transmission curve becomes strongly asymmetric in the high voltage region. This was already expected from eqs. (3a),(4a) whereas eqs. (3b),(4b) are invariant with respect to a transformation \( x \rightarrow -x \). Usually, the modulation transfer function

\[ \text{MFT} = \frac{T_{\text{max}} - T_{\text{min}}}{T_{\text{max}} + T_{\text{min}}} \]  

(7)

is introduced to characterize the resolution capability of the layer. \( T_{\text{min}} \) and \( T_{\text{max}} \) are the minimum and maximum transmittances of the read beam. Fig. 3 shows the MFT vs. grating constant \( g \). The contrast decreases rapidly with decreasing \( g \) and reaches a saturation at large \( g \). The calculations show that the resolution of an SLM improves with decreasing layer thickness \( d \) and large \( \Delta c_1 \). It is optimum at \( K_3/K_{i1} \). For \( \phi = 90^\circ \) the resolution is higher with decreasing \( K_2 \). Resolution can be further improved by raising the surface pretilt angle up to \( \theta_o = 10^\circ \).

![Fig. 2. Time development of transmittance for \( \phi = 90^\circ \) when \( V_i \) is switched from 0 to 0.76V in the high voltage switching region (cf. curve 4 of fig. 1). The parameter gives the time in ms after switching \( V_i \).](image)

![Fig. 3: MTF as a function of the grating g for the 0.01mm cell (notation of the graphs corresponds to that of fig.1).](image)

REFERENCES

Abstract: Numerical modelling has been used to optimise the performance of liquid crystal displays. The modelling and optimisation are described, and the predicted performance is compared with that of actual displays. The accuracy of the numerical model permits its use in LCD fault diagnosis and as a research tool in exploring novel liquid crystal systems.

I. INTRODUCTION

Liquid crystal displays (LCDs) are being used in an ever-increasing variety of applications. Although the bulk of the LCD market is still in simple watch and calculator displays, their use as complex alphanumeric panels for lap-top computers, portable televisions, car and cockpit instrumentation, and in-flight entertainment systems, is growing rapidly. These high added-value products need to be well-tailored to the customer's requirements, so that the ability to optimise their performance has become increasingly important. Whereas the simple watch displays can be designed analytically or by rule-of-thumb, complex displays and those requiring good off-axis performance must be optimised using a numerical model. This paper describes the methods adopted at GEC for this purpose and presents some examples of our results, comparing the predicted performance with that achieved in practice.

II. NUMERICAL MODEL

The modelling problem falls into two parts. First, the LC director profiles, i.e. \( n(z) \), must be calculated for the different display states and, second, the optical properties of these must be calculated. We follow the methods developed by Berreman [1], and find the \( n(z) \) that both satisfies the boundary conditions imposed by the cell construction and the applied voltage, and is a minimum of the total Helmholtz free energy \( F \) where

\[
2F = K_1(n_0(z))\delta^2 + K_2(n_0(z)\cosh z - 2n/P)^2 + K_3(n_0(z)\sinh z)^2 + D_E
\]

(1)

Here the \( K_i \) are the elastic constants and \( P \) the cholesteric pitch of the LC fluid. The device is assumed uniform in the plane of the LC so that \( n \) and \( D \) are functions of \( z \) alone. Then \( D(z) = \varepsilon(z)E(z) = \) constant, and \( \varepsilon(z) \) is calculated from \( n(z) \) and the (uniaxial) dielectric tensor with elements \( \epsilon_{ii} \) and \( \epsilon \). Following Berreman [1], we adopted a shooting method which is interactive, but automatically alerts the user to conditions where the profile is changing rapidly and avoids metastable configurations.

The optical properties of these configurations are calculated using the 4x4 matrix method developed to treat stratified anisotropic planar structures [2]. The field amplitudes at the two surfaces of the structure are related by a propagation matrix \( \hat{P} \) so that

\[
\psi(z) = \hat{P}\psi(0)
\]

(2)

where \( \psi = (E_x, E_y, H_y, -H_x)^T \). \( \hat{P}(z) \) is obtained by writing a differential form of (2) i.e.

\[
\dot{\psi}(z) - i\Omega(z)\psi - D(z)\psi
\]

(3)

in each stratum of the medium, within which the optical properties, contained in \( \Delta(z) \), are constant. Then

\[
P(z) = \exp\left(\int_0^z D(z)dz\right)
\]

(4)

permits the calculation of \( \hat{P} \) from the dielectric tensor elements and orientation as a function of \( z \). Having calculated \( \hat{P} \) it is then straightforward to obtain the transmission and reflection coefficients.

This calculation must be repeated for each wavelength of interest across the visible spectrum, 380-800 nm, and as a function of the angles of the polarisers and any other layers in the device. It must therefore be done efficiently. From (2), \( \hat{P}_{\text{rot}} \) for the entire structure is the product of the \( \hat{P} \)'s for each component, with the appropriate orientational factors. For an LCD with polarisers set at \( (\theta_1, \theta_2) \)

\[
\hat{P}_{\text{rot}} = \hat{P}^{-1}(\theta_2)\hat{P}_{\text{rot}}\hat{P}(\theta_2)\hat{P}_{\text{rot}}\hat{P}(\theta_1)\hat{P}^{-1}(\theta_1)
\]

(5)

where \( \hat{P}_{\text{rot}} \) and \( \hat{P}_{\text{rot}} \) describe the polariser and LC layers respectively and \( \hat{P}(\theta) \) is a rotation matrix. Thus the problem of optimising a display is reduced to performing the matrix multiplication (5) on previously calculated \( \hat{P} \) matrices.

III. DISPLAY OPTIMISATION

The display optimisation process starts by calculating the director profiles in the OFF- and ON-states of the display. For this, we require the elastic constants and permittivities of the LC fluid, the physical parameters of the cell (i.e. thickness, twist and surface tilt), and the drive voltages \( V_{\text{on}} \) and \( V_{\text{off}} \). These voltages are related by the level of multiplexing \( N \) through

\[V_{\text{on}}/V_{\text{off}} = M = \{1/(N+1)\}(N-1)!
\]

and are equally disposed about the central switching voltage \( V_c \) for which the mid-plane tilt is \( 45^\circ \).

From these profiles and the refractive indices of the LC, the \( \hat{P} \) matrices for the LC layer in the two states are calculated and stored for 40 different wavelengths across the visible spectrum. It is essential to include the dispersion of the LC. The \( \hat{P} \) matrix of the polariser is also calculated as a function of wavelength. Then equation (5) is used to calculate \( \hat{P}_{\text{rot}} \) and hence the transmission spectra for the range of polariser angles of interest. The transmission spectra, \( T(\lambda) \), are combined with an illuminant spectrum \( S(\lambda) \) to give the CIE tristimulus co-ordinates \( X, Y, Z \) where

\[X = \int_{\lambda_0}^{\lambda_0+77} S(\lambda)T(\lambda)\gamma(\lambda)d\lambda \]

etc.

(6)

Here \( \gamma(\lambda) \) is the colour matching function and \( \lambda \) is a normalising factor.

\[k = 100/\int_{\lambda_0}^{\lambda_0+77} S(\lambda)\gamma(\lambda)d\lambda \]

Then the tristimulus \( Y \) value is the luminance transmission or "brightness" of the display and \( Y_{\text{off}}/Y_{\text{on}} \) is its contrast ratio. Contour plots of iso-contrast and brightness versus the polar angles enable the optimum display configuration to be selected.

In many cases, a trade-off must be made between contrast and brightness. For the case shown in Fig.1, the polariser angles were...
selected to provide a contrast ratio \( \geq 20 \) with reasonable ON-state transmission when a narrow band green illuminant was used. The predicted transmission spectrum is compared with that measured for the actual display in Fig.1.

![Transmission vs Wavelength](image)

Fig.1 Comparison of calculated (solid line) and measured (broken line) transmission spectra for an optimised SBE display.

IV BLACK AND WHITE SUPERTWIST OPTIMISATION

Supertwist LCDs are inherently coloured, which limits their applicability. The colouration can be removed by including a birefringent film to compensate the optical properties of the LCD. This compensation cannot be ideal, so optimisation is necessary. The method is an extension of equation (5). We write:

\[
P_{\text{film}} = P^{-1}(0_2)P_{\text{pl}}P(0_2)P^{-1}(0_3)P_{\text{pl}}P(0_3)P_{\text{LC}}P^{-1}(0_1)P_{\text{pl}}P(0_1).
\]

where \( 0_2 \) is the orientation of the birefringent layer and \( P_{\text{pl}} \) is its \( P \)-matrix. This can be calculated analytically from its retardation which, in turn, depends on the film thickness. Hence there are now four variables against which to optimise the display: two polariser angles \( 0_1, 0_2 \), the film axis angle \( 0_3 \) and its thickness. The optimisation criterion was taken as the blackest black state along with the whitest white state, so we numerically minimised

\[
\Delta E^2 = (L^* + u^* + v^*)_{\text{dark}} + (L^* - L^*)^2 + u^*2 + v^*2)_{\text{light}}.
\]

![Transmission vs Wavelength](image)

Fig.2 Comparison of calculated (solid line) with measured (broken line) transmission spectra of a film compensated black and white SBE display.

where \( L^*, u^*, v^* \) are the display's co-ordinates in the CIE 1973 uniform colour space and \( L^* \) is the brightness of a pair of parallel polarisers. The optimum values for the parameters were used to construct a display. Fig.2 shows a comparison of the calculated and measured spectra.

V DISCUSSION

The numerical model has proved extremely accurate for LCD design. Besides the examples quoted above, it has been possible to design displays with minimal colour shift between their (reflective)-daylight operation and the (transmissive) backlit mode even though the backlight was a blue-green electroluminescent panel. We have also designed displays for NVG compatibility without using extra filters. It should also be emphasised that the modelling is not confined to normal incidence: Fig.3 shows a calculated iso-contrast plot for an SBE display optimised for two observers at \( \pm 40^\circ \).

![Iso-contrast Plot](image)

Fig.3 Calculated iso-contrast plot for an SBE display optimised for viewing at \( \pm 40^\circ \) incidence. The display is mounted so that the rear LC alignment direction \( (\phi = 0) \) is at \( 45^\circ \).

The modelling programs are sufficiently accurate that realistic simulations, both static and dynamic, of LCD appearance have been implemented [3].

Besides designing displays, the modelling programs can be used to investigate faults in displays. For example, SBE displays are very sensitive to the precise orientation of their polarisers. It is impossible to measure these angles after cell assembly, but by measuring the display performance (i.e. its contrast ratio and transmission) the polariser orientation can be accurately deduced from a comparison with the model predictions. Thus faults in the display fabrication can be readily identified.

Finally, it should be noted that these programs are an invaluable research tool. In ferroelectric LCDs, the director profile is generally unknown. By measuring their optical properties and comparing these with the predicted characteristics of plausible models, a predictive capability for FLCDs can also be developed. In conclusion, the numerical modelling programs for LCDs have proved to be very accurate. They have been used to design custom displays, diagnose faults and as a tool in display research.

REFERENCES

CONTINUUM THEORY OF SMECTIC C LIQUID CRYSTALS;
STATIC CONFIGURATIONS AND ELEMENTARY FREDERIKS TRANSITIONS

I. W. STEWART
Department of Theoretical Mechanics,
University of Nottingham,
University Park,
Nottingham, NG7 2RD, U.K.

F.M. LESLIE and T. CARLSSON
Mathematics Department, Livingstone Tower,
Strathclyde University,
26 Richmond Street,
Glasgow, G1 1XH, U.K.

ABSTRACT — Smectic C liquid crystals are shown to exhibit six types of static configuration which satisfy the equilibrium equations of the continuum theory introduced in [1]. A brief mathematical description of the resulting families of parallel surfaces which correspond to these static solutions is given. A possible configuration is a wedge of concentric cylinders, a Frederiks transition induced by an appropriate electric field is discussed in this case.

I. INTRODUCTION

To describe the layer structure of a smectic C liquid crystal we introduce the unit layer normal \( \mathbf{a} \) which is subject to the constraint [2]:

\[
\nabla \times \mathbf{a} = 0 .
\]

As in [3], a unit vector \( \mathbf{c} \) which is perpendicular to \( \mathbf{a} \) is used to describe the direction of tilt of the alignment with respect to the layer normal. The two directors \( \mathbf{a} \) and \( \mathbf{c} \) satisfy the constraints

\[
\mathbf{a} \cdot \mathbf{a} = c \mathbf{c} = 1 , \quad \mathbf{a} \cdot \mathbf{c} = 0 .
\]

The energy integrand used in smectic C theory is

\[
2W = K_1 (\nabla \mathbf{a} \cdot \nabla \mathbf{a})^2 + K_2 (\nabla \mathbf{c} \cdot \nabla \mathbf{c})^2 + K_3 (\mathbf{a} \cdot \nabla \mathbf{c})^2 + (\mathbf{c} \cdot \nabla \mathbf{a})^2
\]

\[
+ 2 K_4 (\nabla \mathbf{a} \cdot \nabla \mathbf{c}) (\mathbf{a} \cdot \nabla \mathbf{c}) + 2 K_5 (\mathbf{a} \cdot \nabla \mathbf{c}) \mathbf{c} + 2 K_6 (\nabla \mathbf{a} \cdot \mathbf{c}) \mathbf{a}
\]

where the \( K_i \) are elastic constants (see [1],[4],[5]). The corresponding equilibrium equations examined are (see [1],[6])

\[
\frac{\partial w}{\partial \mathbf{a}_1} , \mathbf{J} = - \frac{\partial w}{\partial \mathbf{a}_1} + G_{\mathbf{a}_1} + \lambda_1 + \mu_1 + \phi_1 \mathbf{J} \mathbf{a}_1 , \mathbf{J} = 0
\]

\[
\frac{\partial w}{\partial \mathbf{c}_1} , \mathbf{J} = - \frac{\partial w}{\partial \mathbf{c}_1} + G_{\mathbf{c}_1} + \mu_1 + \phi_1 = 0
\]

where \( \phi, \lambda, \mu \) and \( \kappa \) are Lagrange multipliers arising from the constraints (1) and (2) and \( \phi \) is the usual alternator. Solutions to (4) provide static configurations for smectic C liquid crystals.

II. STATIC SOLUTIONS

There are six types of well behaved surface which readily provide static solutions for a restricted six term version \( (K_1-K_6) \) of the energy given in (3), namely, the Dupin cyclides, circular tori of revolution, spheres, parabolic cyclides, infinite cylinders and planes (see [1],[5]-[7]). Solutions to equations (4) for \( \mathbf{a} \) and \( \mathbf{c} \), including any related Lagrange multipliers, can be found and as an example we now mention the parabolic cyclides [7].

The Cartesian equation of a parabolic cyclide is [7]

\[
x(x^2+y^2+z^2) = \mu (x^2+y^2) - z^2 (2+\mu) - (x-\mu z)^2 (2+\mu)^2 = 0
\]

where the confocal parabolae essential to its construction are

\[
y^2 = 40 (x^2+\delta) \quad z^2 = -40 \delta \quad z = 0 \quad y = 0
\]

and \( \ell \) and \( \mu \) are real parameters. Equation (5) may be parameterized as

\[
x(1+\delta z^2+\mu t) = \mu (t^2+1) + \ell (t^2-\delta^2-1)
\]

\[
y(1+\delta z^2+\mu t) = 2t (\ell (t^2+1)+\mu)
\]

\[
z(1+\delta z^2+\mu t) = 2t (\ell t^2+\mu)
\]

where \( \ell < \mu, \delta, t < +\infty \) (varying \( \delta \) and \( t \) while keeping \( \mu \) fixed) gives one particular parabolic cyclide surface. Transforming from the Cartesian to the \((\mu,\delta,t)\) frame we can show that

\[
\mathbf{a} = (1,0,0) \quad \mathbf{c} = (0,1,0)
\]

satisfy the constraints (1) and (2) and solve the transformed version of (4) upon a suitable choice of Lagrange multipliers. The consequences of varying \( \ell \) and \( \mu \) are discussed in [7].

III. FREDERIKS TRANSITION IN A WEDGE

![Fig. 1](image)

The smectic layers are assumed to be parts of concentric cylinders with the common axis coinciding with the centre of the wedge. The layer normal \( \mathbf{a} \) is in the \( \ell \) direction and the angle between the boundary plates is \( \beta \).

Introducing the vector

\[
\mathbf{b} = \mathbf{a} \times \mathbf{c}
\]

we can rewrite the energy integrand as (see [4],[8])

\[
2W = A_1 (\mathbf{b} \cdot \nabla \mathbf{c})^2 + A_2 (\mathbf{c} \cdot \nabla \mathbf{c})^2 + 2 A_3 (\mathbf{c} \cdot \nabla \mathbf{b}) (\mathbf{c} \cdot \nabla \mathbf{b})
\]

\[
+ B_1 (\mathbf{b} \cdot \nabla \mathbf{c})^2 + B_2 (\mathbf{c} \cdot \nabla \mathbf{b})^2 + B_3 (\mathbf{b} \cdot \nabla \mathbf{b} \cdot \nabla \mathbf{c})^2 + 2 B_4 (\mathbf{b} \cdot \nabla \mathbf{b} \cdot \nabla \mathbf{c}) + 2 C_1 (\mathbf{c} \cdot \nabla \mathbf{c}) (\mathbf{b} \cdot \nabla \mathbf{c})
\]

\[
+ 2 C_2 (\mathbf{c} \cdot \nabla \mathbf{c}) (\mathbf{b} \cdot \nabla \mathbf{b})
\]

where the \( A_i, B_i \) and \( C_i \) are elastic constants. Introducing the cylindrical coordinate system \((\rho,\phi,z)\) we set

\[
\mathbf{a} = \hat{\rho}
\]

\[
\mathbf{b} = -\hat{\rho} \cos \phi + \hat{\phi} \sin \phi
\]

\[
\mathbf{c} = \hat{\phi} \sin \phi + \hat{\rho} \cos \phi
\]

and assume that \( \varphi = \varphi(0) \) with \( \varphi(0) = \varphi(\beta) = 0 \) where \( \beta \) is the wedge angle (see Fig. 1). To induce a Frederiks transition we apply an electric field of the form

\[
\mathbf{E} = \mathbf{U} \sin \phi
\]

across the bounding plates of the wedge of concentric cylinders where \( \mathbf{U} \) is the voltage. The corresponding electrical energy integral is

\[
2W_0 = -e \mathbf{d} \mathbf{E} \mathbf{U} \sin^2 \phi \sin^2 \varphi
\]

where \( e \mathbf{d} \) is the dielectric anisotropy of the liquid crystal, \( e \mathbf{d} \) is the permittivity of free space and \( \varphi \) is the usual fixed tilt angle associated with smectic C liquid crystals. To obtain a threshold for a Frederiks transition we add the integrands (10) and (13) and minimize the integral of this sum with respect to \( \alpha \) over the
region $r_1 < r < r_2$, $0 < \alpha < \beta$, $z_1 < z < z_2$. This is achieved by solving the following linearized Euler-Lagrange equation (see [8])

$$
\tilde{B}_2 \frac{d^2 \varphi}{d z^2} + 2(\overline{A}_{21} + \overline{A}_{11}) \varphi + \varepsilon_n \varepsilon_0 \left( \frac{\partial}{\partial z} \right)^2 \varphi = 0
$$

(14)

where $\tilde{B}_2$, $\overline{A}_{21}$, and $\overline{A}_{11}$ are the suitably adjusted temperature independent parts of the corresponding original elastic constants appearing in (10). To satisfy $\varphi = 0$ at $\alpha = 0, \beta$ we make the ansatz

$$
\varphi(\alpha) = \varphi_m \sin \left( \frac{\pi \alpha}{\beta} \right)
$$

(15)

where $\varphi_m$ is suitably small. Inserting this ansatz into equation (14) we derive the Fredericks threshold $U_C$ as

$$
\varepsilon_n \varepsilon_0 \varphi_m^2 = \pi^2 \tilde{B}_2^2 + 2 \beta^2 (\overline{A}_{21} + \overline{A}_{11})
$$

We therefore see that by varying the wedge angle $\beta$, we can evaluate the constants $\tilde{B}_2$ and $(\overline{A}_{21} + \overline{A}_{11})$ by measuring the Fredericks threshold for each value of $\beta$.

REFERENCES

OPTICAL TRANSMISSION MODELLING OF FERROELECTRIC LCDs


Dept. Tecnología Fotónica, ETSI Telecomunicación, Ciudad Universitaria
28040 Madrid Spain

Abstract: Modelling of ferroelectric liquid crystal displays involves angular and spectral transmission of real backlights, using commercially available polarisers and actual FLC mixtures. Running times for a complete case may be extremely large. Therefore optimised procedures and simplified models for precalculations become necessary. A fairly accurate description of the optical behavior of real FLC displays in working conditions has been achieved. Polariser angles optimisation, spectral and angular transmission and color coordinates calculations are the most useful results.

I. INTRODUCTION

The use of ferroelectric liquid crystals (FLC) for manufacturing high quality displays has drawn considerable attention since surface stabilized bistable structures of these materials were reported by Clark and Lagerwall [1]. A simple bookshelf geometry with uniform layers perpendicular to the glass plates was proposed, and a fast electrooptical switching was demonstrated. Further work by them and other researchers [2-4], however, showed that the usual configurations found in real structures are far more complicated, tilted bookshelf, splayed or chevron.

Optical transmission models may greatly help to develop these devices. Under the point of view of commercial displays, the most interesting information that a computer model may provide is the one related to optical transmission properties of actual devices in working conditions.

II. PROBLEM FORMULATION

Two models for angular and spectral distribution of light transmitted through FLC samples have been prepared and tested. The first model -called slab model- can compute light transmission through FLC cells whose director orientations are defined by a small number of homogeneous slabs. The mathematical description of the optical system (i.e., polariser, glass, anisotropic material, glass, and polariser) is based on geometrical optics [5].

The optical transmission calculations are performed for all possible angles (0-360°, 10° steps) and oblique incidences (0-80°, 5° steps). Driven and zero volts states are computed by suitably orienting the liquid crystal slab directors within the optical system. Data for real polarisers and backlights (380-770 nm, every 10 nm) are then included, and the overall transmission for every angle is computed.

The second, so-called continuous-varying profile (CVP) model, is based on Berreman 4x4 formulation [6,7] and may be used for any arbitrary director profile. Input data and output results are defined as above. The truncated series expansion of Berreman’s method is avoided using explicit expressions for 4x4 propagation matrices of homogeneous slices of the material, as proposed by Wöhlc et al. ‘8]. Results from both methods are similar, the former being faster but restricted to simplest profiles.

III. RESULTS

In the examples given below, the following elements are used:
- Backlight: C.I.E. D65
- Color filters. EMI RGB/standard fluorescent lamp
- Polarizers: G1220
- Material: ZLI 3654 (Merck).

![Figure 1.- Transmission spectra for different thicknesses. Backlight: D65. Slab model.](image-url)
The high birefringence of the clear state in FLC displays may influence the output of wide spectrum backlights. Figure 1 shows the spectral response of a 15° tilted bookshelf with a 3° surface pretilt angle. The region of shorter wavelengths is remarkable for higher thicknesses. The clear state is behaving like an almost ideal retardation plate. The models predict little spectral changes for samples below 2μm for a typical birefringence. This should be considered a design parameter, specially in colour displays, where luminance of the blue filter may be considerably reduced.

Figure 2 is a 3D plot of oblique incidences. A 15° chevron with 10° pretilt at 30 °C is represented. The central region of the plot corresponds to nearly normal incidences; angles are linearly varying with distance from the centre. The outer regions are grazing incidences up to 80°. Rotation of the viewing angle is achieved by rotating the figure. While the clear state yields a fairly symmetric hat shape, the dark state shows small maxima at 45° for nearly grazing incidence. This is due to residual light passing through non-ideal polarisers. Contrast curves may be directly derived from these plots.

Figure 3 shows the evolution of dominant colours and saturation when sample thickness is varied from 0 to 5μm. No other cell parameter (tilt angle, pretilt, birefringence, FLC configuration, oblique incidence, chevron/tilted bookshelf angle) produces any relevant colour migration. However, sample thickness (strictly, the product Δn.d) produces dramatic effects in colour coordinates as it does in luminance (Fig. 1). Again, thicknesses below 2μm are advisable. The central curve of figure 3 shows a D65. The outer curves are red, green and blue filters currently used for commercial colour displays.

An experimental setup consisting of a He-Ne laser, two fixed crossed polarisers and a FLC sandwich twisted 45° has been prepared to test oblique incidences. The sample is rotated 360° while switching. The output in polar coordinates (Fig. 4) is a daisy-shaped curve where four "petals" are maxima from one of the states interlaced with another four maxima from the other state. Both theoretical and experimental data are normalised using normal incidence. A fair match of the results is found, specially in the angular values of the maxima. These values are related to the projection of the tilt angle on the plane of incidence thus giving information of the actual FLC configuration.

REFERENCES

USING SIMULATED ANNEALING IN THE DESIGN
OF COVERING CODES

PATRIC R. J. ÖSTSTORGÅRD
Digital Systems Laboratory
Helsinki University of Technology
02150 Espoo, FINLAND

Abstract
Simulated annealing has turned out to be a very efficient method in the search for covering codes. We discuss three methods, all using simulated annealing, for finding such codes. The direct approach and the approach using matrices have earlier been applied to some extent. The third method, the use of simulated annealing in finding acceptable partitions for seminormal codes, is novel. All these methods have led to new codes, better than any other known in the literature.

I. INTRODUCTION
Covering codes originated back in 1918, when Taussky and Todd [10] discussed the problem group-theoretically. During the years many contributions to the research have appeared in different journals within the areas of coding theory and combinatorics. The main interest has concerned binary codes and ternary codes of covering radius 1 (the so called football pool problem, see e.g. [8, 11]). There has been considerable recent growth in the interest in the area (cf. [3]).

The problem of finding covering codes can be formulated as a combinatorial optimization problem. Simulated annealing (SA) has turned out to perform amazingly well in the search for such codes.

II. COVERING CODES
We consider the space \( F_q^n \) consisting of all words of length \( n \) and coordinates belonging to the set \( \{0, \ldots, q - 1\} \). A code \( C \subseteq F_q^n \) covers \( F_q^n \) with radius \( R \) if every word in \( F_q^n \) is within Hamming distance \( R \) from some codeword in \( C \). The Hamming distance between two words is the number of coordinates in which they differ. A \( q \)-ary code of length \( n \) that has \( M \) codewords and covering radius \( R \) is called a \((q, n, M)R\) code. We denote \( K_q(n, R) = \min \{M \mid \text{there is a } (q, n, M)R \text{ code}\} \).

Example 1. \( F_2^3 = \{000, 001, 010, 011, 100, 101, 110, 111\} \). The code \( C = \{000, 111\} \) is a \((2, 3, 2)\) code, since the words within Hamming distance 1 from 000 are \( C_1 = \{000, 001, 010, 100, 110\} \) and from 111 are \( C_2 = \{111, 110, 101, 111\} \), and \( C_1 \cup C_2 = F_2^3 \). \( K_2(3, 1) = 2 \), since there is no \( (2, 3, 1) \) code as is easily seen.

Codes with different kinds of partitioning properties have turned out to be very useful in constructing new codes from old ones. The concepts of normal and seminormal codes were introduced in [3] and [4], respectively, and in [13] the notion of seminormal codes was presented. In this paper we show have simulated annealing can be adopted to prove the seminormality of codes. We now define \( k \)-seminormal and strongly \( k \)-seminormal codes.

Definition 1: (Östergård, [13, Definition 2]) A \((q, n, M)R\) code \( C \) is said to be \( k \)-seminormal, if there is a partition of \( C \) into \( k \) subsets \( C_0, \ldots, C_k \) such that, for all \( x \in F_q^n \), with \( d(x, C) = R \),

\[ \max \{d(x, C_i)\} \leq R + 1. \]

Such partitions are called acceptable. The definition of strongly \( k \)-seminormal codes is obtained by removing "with \( d(x, C) = R \)" from Definition 1. The following theorem shows how seminormal codes can be used in the construction of new covering codes.

Theorem 1: (Östergård, [13, Theorem 3]) If there is a \((q, n, M)R\) seminormal code and \( q \) is a prime power, then there is a \((q, n + q, q^2 - 2M)R + 1 \) code.

Strongly seminormal codes are also of importance in constructing covering codes. These constructions are not presented here, but the interested reader is referred to [13].

III. SIMULATED ANNEALING
The algorithm used in our research is only a slight modification of the original simulated annealing algorithm presented by Kirkpatrick et al. [6, 7].

In the presentation of the methods that have been used we stress presentations on the definition of the neighbouring configurations and the energy function. Appropriate cooling schedules can usually be determined by some practical experiments.

IV. THE METHODS
Although we try to minimize the number of codewords covering a certain space (to determine \( K_q(n, R) \)), the methods to be presented here attempt to find coverings with a certain number of codewords. The finalization is attained by sequent attempts (runs) to find coverings with fewer and fewer codewords (the direct approach).

A. The Direct Approach
The first results concerning the use of simulated annealing in coding theory were presented by El Gamal et al. in 1987 [2]. In that paper covering codes were not considered, however, later the same year these were treated by Wille in [11].

The implementation of the direct approach is quite straightforward. The energy function is simply the number of words not covered by the codewords. When we are searching for a code \( C \) that covers \( F_q^n \) with radius \( R \) the energy function is

\[ E = |\{x \in F_q^n \mid d(x, C) > R\}|. \]

In the annealing process we go through all codewords. A neighbourhood configuration is obtained by replacing a codeword \( c \) with a randomly chosen word \( c' \), such that \( d(c, c') \geq R \). The codewords could also be replaced with completely random words, however, experiments have showed that this way of defining the neighbourhood configuration is inferior to the previously mentioned one.

It has turned out that for covering radius \( R = 1, 2, 3 \) codes with up to about 100, 40 and 15 codewords, respectively, can be found within reasonable time. Even an ordinary personnel computer can be used for this (most of our work has been done on a 10 MHz PC/AT-compatible computer). With an increasing number of codewords the processing time required grows very fast, and the search for the \((3, 7, 180)\) code found in [8] required many attempts and an extremely slow cooling rate. In the binary case the following theorem can be used to overcome this problem in some cases.

Theorem 2: (Östergård, [12, Theorem 2]) Let \( C \subseteq F_q^2F_q^2 \) be a code of covering radius \( R \) that has \( n \) codewords. Then there is a \((2, 6 + 3q, 2^n)R \) code.

In the following subsections two other methods for finding coverings with many codewords are discussed.
B. Covering Using Matrices

The method to be presented in this section was first considered by Kamps and van Lint [5] and later generalized by Blokhuis and Lam [1]. In [9] van Lint Jr. generalized it to arbitrary covering radii.

Let \( A = (I; M) = (a_1, \ldots, a_n) \) be a \( r \times n \) matrix where \( I \) is the \( r \times r \) identity matrix and \( M \) is a \( r \times (n-r) \) matrix with entries from \( F_q \). For \( s \in F_q^r \) the \textit{R-covering of} \( s \) using \( A \), \( S_{A,R}(s) \), is defined as

\[
S_{A,R}(s) = \{ s + \sum_{j=1}^{n} \alpha_j a_j | \alpha_j \in F_q, (\alpha_j \neq 0) \leq R, 1 \leq j \leq n \}.
\]

Consequently, \( A = I \) corresponds to covering in the traditional sense. A subset \( S \) of \( F_q^r \) \textit{R-covers} \( F_q^r \) using \( A \) if

\[
F_q^r = \bigcup_{s \in S} S_{A,R}(s).
\]

**Theorem 3:** (van Lint Jr., [9, Theorem 1.4.4]) If \( S \) \textit{R-covers} \( F_q^r \) using an \( r \times n \) matrix \( A = (I; M) \), then \( W = \{ w \in F_q^n | Aw \in S \} \) covers \( F_q^n \) with radius \( R \). \( |W| = |S| \cdot q^{n-r} \).

The optimization problem now consists of finding the words in \( S \) and the \( M \) part of the matrix \( A \). The interconnection between the codewords and the matrix causes problems, and it is not immediately clear how to perform the search efficiently. We consider three different ways to deal with this problem.

1. If there is a small number of nonequivalent matrices \( M \) it is possible to do the annealing in trying to find a set \( S \) for all these possibilities. This is the approach in [14]. A considerable speed-up can be achieved by performing the first run with a fast cooling rate, after which values of \( M \) leading to bad coverings can be removed. Repeating this procedure with slower and slower cooling rates and less and less numbers of values of \( M \) possibly leads to a covering. This approach has turned out to perform very well, however, it cannot be applied when there is a big number of nonequivalent matrices \( M \).

2. In the annealing process all words in \( S \) are gone through in one round (like in the direct approach), and after that attempts are made to change the columns of \( M \) with other columns. This is the approach in [8]. The performance of this approach has not turned out to be very satisfactory.

3. We now describe how simulated annealing can be used on two levels to attack the problem. We first try to find a good value of \( M \). For a certain \( M \) we try to find a good covering by searching for a set \( S \). This is done using simulated annealing. Neighbourhood configurations are now obtained by replacing the columns of \( M \) with columns that differ in one position. These changes are accepted in the normal way depending on the number of uncovered words remaining after the next cooling. Having fixed \( M \) attempts are made to determine the minimal size of \( S \).

C. Partitioning Seminormal Codes

We consider a \( (q,n,M) \) code \( C \) and describe a procedure that can be used in trying to prove \( k \)-seminormality of the code. An acceptable partition is found at the same time.

We divide the \( M \) codewords (preferably randomly) into \( k \) sets. If \( k \) divides \( M \), the size of the sets is \( M/k \), otherwise the size of some sets is \( [M/k] \) and of some sets \( [M/k] \). These sets are named \( C_0, \ldots, C_{k-1} \).

The energy function can now be taken as

\[
E = |\{(x, s) \in (F_q^r, \{0, \ldots, k-1\}) | d(x, C_s) > R+1, d(x, C) = R\}|.
\]

In the annealing process we go through all the codewords of all the sets and the random displacement consists of changing the codeword in question and a random codeword in another set.

If we want to prove strong \( k \)-seminormality, only a slight change to the energy function has to be made.

V. CONCLUSIONS

Simulated annealing has turned out to perform surprisingly well in the search for covering codes, as a matter of fact probably no other area of coding theory has profited so much by this method. This can partly be explained by the fact that for example packing (error-correcting) codes are very structural in their nature and so for instance algebraic methods can be applied in the constructions. However, many known record-breaking coverings (the \((3,6,73)\) code found in [8] to mention one) seem to possess no pattern at all.

In spite of the good results reported here, there is still some work to be done on this subject. This concerns especially the method of covering using matrices, with its special problems.

ACKNOWLEDGMENT

This work has been financially supported by the Finnish Academy.

REFERENCES


A GENERAL PURPOSE DISTRIBUTED IMPLEMENTATION OF SIMULATED ANNEALING

R. Diekmann, R. Lüling, J. Simon
Department of Mathematics and Computer Science
University of Paderborn, Germany

Abstract: We present a problem independent general purpose parallel implementation of simulated annealing on distributed message-passing multiprocessor systems. We give a classification of combinatorial optimization problems. For typical representatives of the different classes good parallel simulated annealing implementations are presented. A new parallel SA-implementation is introduced. It works simultaneously on several markov chains and decreases the number of chains dynamically.

I. Introduction
Simulated Annealing (SA) was first presented by Kirkpatrick et al. [5] for solving hard combinatorial optimization problems and has proven to be a good technique to a lot of applications [3,4,6]. The disadvantage of this probabilistic approach is its large amount of computation time needed for obtaining a near-optimal solution. Some attempts at speeding up simulated annealing by using parallel computing systems have been made.

Two parallelization strategies are possible. In the first, problem describing data is distributed among several processors. This kind of parallelization depends on the problem and is not easy to find for every optimization problem. Therefore we consider a second strategy which is based on the parallelization of the actual simulated annealing algorithm. Every processor gets the complete problem instance and executes the sequential steps of the annealing algorithm in parallel. This approach is problem independent and easy to adapt. Some authors describe this technique using parallel systems with shared memory [1,3]. For special problems [2] distributed systems are used in the same way.

Aarts et al. [1] first described a technique where all processors work in parallel on the evaluation of one markov chain. With this idea we obtain a speedup of 23 with 64 processors when using a selfadapting cooling schedule.

We present an improved parallel SA algorithm that provides a speedup of 41 on 121 processors using a selfadapting cooling schedule. In this approach all processors start working on a number of distinct markov chains. The number of chains is reduced dynamically until all processors evaluate only one chain.

All algorithms are implemented in OCCAM-2 on a fully recon-figurable transputer system.

II. Classification
The optimization problems are classified according to their evaluation time for the different steps of the sequential annealing algorithm. The most important two classes are:
1. problems where the generation of new configurations has an equal time (e.g. TSP).
2. problems where the cost evaluation is much more expensive than the generation of configurations (e.g. placement).

III. Parallelization
We examined several parallelization approaches known from literature concerning their suitability for problems of the different classes. Kirkpatrick's ideas of locking parts of the configurations [3] were found to be inefficient when implemented in large distributed systems because no fast locking-mechanism is available.

Baiardi's model of a processor farm [2] provides nearly optimal speedup when used for problems of the second class. In this approach new configurations are generated by one master processor and sent to a number of slave processors for cost evaluation and acceptance decision. If this method is applied to problems of the first class, the master is a bottleneck. He is not able to generate enough new configurations to keep a large number of slaves working.

III.a One-Chain
To get rid of this bottleneck new configurations must be generated by the slaves themselves. Only system updates are still controlled by the master. A version of this idea was described by Aarts et al. and implemented on a small parallel machine with shared memory [1].

In our implementation all slaves work on the evaluation of one markov chain. If a slave detects an acceptable configuration the master is informed. He initiates a system update. For convergence properties it is not profitable to use the first detected acceptable configuration for a global update. In a synchronization phase the master collects all accepted configurations and chooses one for a global update. Quickly calculated transitions are not favored.

Fig.1: Results of One-Chain with a selfadapting cooling schedule

A ternary tree is used as hardware topology for our implementation. It provides minimal path length from the master-node (root of the tree) to all other nodes and has a maximal degree of four (since each processor has 4 communication links).

Using a fixed cooling schedule (with fixed initial temperature $T_0$, temperature decrement $T_n = \alpha T_{n-1}$ and fixed chain length $L$) we achieve nearly linear speedup results.

A selfadapting cooling schedule according to Huang et al. [4] is implemented. For the evaluation of the standard deviation $\rho$ we use a smoothing technique similar to Otten [6].
Measurements of speedup and efficiency are shown in fig. 1. With this technique, it is not profitable to use more than 40 processors. The convergence behavior is equal to that of the sequential algorithm.

III.b Par-Chain

The behavior of the One-Chain is worse in the presence of high temperatures when implemented in large distributed systems. Many synchronization phases appear because almost every new configuration is accepted. For that reason no linear speedup is achieved.

To overcome this disadvantage several markov chains can be used and speedup is achieved by shortening the length of the individual chains. To guarantee convergence the chain length cannot be reduced arbitrarily. Since at lower temperatures the behavior of the One-Chain is much better, the idea was to combine these two methods [1].

When the algorithm starts all processors calculate their own markov chain according to the One-Chain method (fig. 2). After a certain time a global synchronization step is made. All processors send their configurations to the root-node which controls the masters of One-Chain. It chooses one of the configurations for further work.

In case of low acceptance-rates, processors are clustered dynamically. This is done by building subtrees (fig. 3). Each cluster now calculates one chain according to the One-Chain implementation. While the number of chains is decreased, the chain length is magnified. The combination steps are repeated until all processors compute only one markov chain (fig. 2).

The cooling schedule is controlled by the root-node. Calculation of initial temperature and temperature reduction are done in the same way as in the One-Chain implementation. The equilibrium detection cannot be done depending on the values of accepted configurations (like in One-Chain) because not all values from all clusters are available on the root-node. Therefore L is calculated before the computation of a chain starts according to a technique used by Otten [6]. Measurements of speedup and efficiency can be seen in fig. 4.

IV. Conclusion and further work

We presented a general purpose parallel implementation of SA on distributed multi-processor systems. This approach is problem independent and easy to adapt. Hence it can be used as a tool of universal application for approximation of hard combinatorial optimization problems. To our knowledge it is the first problem independent parallelization of SA for large distributed systems.

Our next step is to combine methods from genetic algorithms [7] with the given SA parallelizations. Better convergence behavior is achieved by using a pool of markov chains. In the global synchronization phase of the Par-Chain algorithm the root-node chooses a pool of configurations for further work. The members of the pool and their rate of reproduction are selected concerning their 'goodness'. First experiments indicate also a better behavior in speedup.

References

Parallelizing SA For Graph Embedding Is Hard *  

John E. Savage  
Department of Computer Science, Box 1910  
Brown University  
Providence, RI 02912  
jes@cs.brown.edu

Markus G. Wloka  
Department of Computer Science, Box 1910  
Brown University  
Providence, RI 02912  
mgw@cs.brown.edu

Abstract: We show that local search heuristics for grid and hypercube embeddings based on the successive swapping of pairs of vertices, such as simulated annealing, are P-hard and unlikely to run in polylogarithmic time. We have developed and implemented on the Connection Machine CM-2 a new massively parallel heuristic for such embeddings, called the Mob heuristic, which gives excellent results in practice.

1 Introduction

Graph embedding is the NP-complete problem of mapping a graph into another graph, called a network for clarity, while minimizing a cost function on the embedded edges of the graph. Graph embedding has application in VLSI (Very Large Scale Integration) placement and the minimization of data movement in parallel computers. An automatic graph-embedding tool optimizes communication resources, permits fault tolerance, and allows parallel programs to be divorced to some extent from the structure of the underlying communication network.

In this paper, we address grid and hypercube embeddings. The cost of a grid embedding is the sum of the half-perimeters of the boxes enclosing each edge. The cost of a hypercube embedding is the sum of the Hamming distances between the two vertices of each edge. Neither cost function measures routing congestion, but reduction of edge lengths can reduce congestion as a secondary effect.

Local search heuristics, of which steepest descent, Kernighan-Lin [2], and simulated annealing [3] are well-known examples, have been established as the heuristics of choice for general graph-embedding problems. The recent availability of general-purpose parallel processing hardware and the need to solve very large problem instances have led to increasing interest in parallelizing local search heuristics.

In local search algorithms, an initial solution is constructed, usually by some random procedure, and the cost function $f$ is computed. Changes are made to the current solution, and the new solution, which we say is in the neighborhood of the current solution, replaces the current solution. The SWAP neighborhood of an embedding is the set of embeddings obtained by swapping the embedding of two vertices.

2 P-hardness Results

We show that local search algorithms for grid and hypercube embeddings are expected to be hard to parallelize by reducing them to the circuit value problem (CVP), the canonical P-complete problem.

Problems in NC can be solved on a parallel machine with polynomially many processors in polylogarithmic time. If a P-complete problem is in NC, then P, the class of problems solvable in polynomial time, is contained in NC, a highly unlikely result. Since logspace-reducibility is transitive, if a problem $A$ is P-complete and we can find a logspace reduction of it to another problem $B$ in P, then $B$ is also P-complete. The definition of P-hardness does not require that the decision problem be in P. A P-hard problem is at least as hard to solve as a P-complete problem.

CVP is the problem of computing the value of a Boolean circuit from a description of the circuit and values for its inputs. CVP is P-complete [4]. Monotone CVP, a restricted version of CVP which uses only the operations AND and OR, is also P-complete [1].

Graph partitioning is a special case of graph embedding. The graph-partitioning problem is to partition the vertices of an undirected graph $G = (V, E)$ ($|V|$ even) into two sets of equal size such that the number of edges between them is minimized. We use the following result to give reductions to local search algorithms for grid and hypercube embeddings.

Theorem 1: For the graph-partitioning problem, local search under the Kernighan-Lin neighborhood is P-complete. Local search under the SWAP neighborhood is P-complete, or P-hard if the local search algorithm is randomized [5, 7].

The result that local search under the SWAP neighborhood for graph partitioning is P-complete was independently obtained by Yannakakis and Schäffer [9].
The graph-partitioning completeness proof can be summarized as follows. A graph $G$ is constructed from a monotone boolean circuit $C$. The graph $G$ contains subgraphs corresponding to the AND and OR gates in the circuit $C$ as well as auxiliary subgraphs. A partition of $G$ into equal-sized sets is given as a starting point. By applying improving swaps until a local minimum is reached, the value of the circuit $C$ can be computed directly from the resulting graph partition.

Theorem 2 For the graph-embedding problem on the grid and hypercube, local search under the SWAP neighborhood is P-complete, or P-hard if the local search algorithm is randomized.

To prove Theorem 2, we give two logspace reductions that map the graph $G$ used in the graph-partitioning proof into initial embeddings i.e. the grid and hypercube. The full description of the constructions are given in [6]. We give a sketch here of the grid embedding construction, the construction for the hypercube embedding problem is similar.

For the grid embedding problem, the construction shown in Figure 1 consists of a grid with two vertical lines $L_0, L_1$ which have unit horizontal separation. The line $L_0$ represents a logical value of 0 while the other line represents a logical value of 1. Let $(X_0, Y_0)$ be the initial partition of $G$. Let $K$ be the maximum degree of $G$. $(K - 9)[5]$. The vertices in the set $X_0$ of $G$ are placed on $L_0$ as shown in Figure 1 and above them is an equal number of solitary vertices. The vertices in $Y_0$ are placed on $L_1$ about an equal number of solitary vertices. These vertices on the vertical lines $L_0$ and $L_1$ are spaced $D$ vertical grid points apart, where $D = K + 1$, to leave enough room for the rest of the construction. With this construction we show that the only swaps accepted by the local search heuristic are between a vertex of $G$ and the opposing solitary vertex.

To limit the movement of a vertex $v$ in $G$, we place $L$ ($L = 2K + 2$) $D 	imes D$ anchor cliques to the left and another $L$ anchor cliques to the right of the two vertical lines, and we connect the closest vertex of every clique to $v$ with an expander edge, of which there are $2L$. The solitary vertices are not anchored, and can thus move freely. We shall show that the expander edges and cliques constrain $v$ to move only along the $z$-axis between the two vertical lines.

Both constructions have the property that all possible positive gain swaps in the problem correspond to positive gain swaps in the associated graph-partitioning problem, and thus mimic the computation of the value of a circuit. We find that any local-search based algorithm is P-hard if every swap accepted by it corresponds to a positive gain swap in the graph-partitioning algorithm.

Since local search heuristics based on the SWAP neighborhood for grid and hypercube embeddings are P-hard, it is unlikely that a parallel algorithm exists that can find even a local minimum solution in polylogarithmic time in the worst case. This result puts experimental results reported in the literature into perspective: attempts to construct the exact parallel equivalent of serial simulated annealing based heuristics for graph embedding have yielded disappointing parallel speedups.

3 The Mob Heuristic

We have developed a new massively parallel heuristic, which we call the Mob heuristic. The heuristic is closely related to both Kernighan-Lin and simulated annealing. The algorithm uses a mob-selection rule to swap large sets of vertices, called mobs, across planes of a grid or hypercube. If the new embedding found has a smaller cost, the search is repeated on the new embedding with the same mob size. If the cost increases, the neighborhood of the new embedding is searched with a smaller mob size. We assume that Mob executes a number of iterations that does not exceed a polynomial in the number of graph vertices. A "schedule" determines the rate at which the mob size decreases. The P-hardness result given above applies also to the Mob heuristic if the mob size is fixed at one.

The mob-selection rule searches for an approximation to the subset that causes the largest improvement in the embedding cost and is designed to be computed very quickly in parallel. On the hypercube a hyperplane is chosen at random and vertices are swapped between hypercube neighbors across the chosen hypercube axis. On the grid, a distance $D$ of 1, 2, 4, 8, 16, ... on either the $x$ or $y$ axis is chosen at random, and vertices are swapped between grid neighbors that are a distance $D$ apart. In both cases a gain is computed for every vertex to find the vertex pairs whose exchange would cause the largest individual change in the embedding cost. A group of high-gain vertices of size mob is selected to move. To avoid sorting by gain, an expensive operation on a massively parallel machine, we use adaptive binary search to identify a set of vertices with gain $g$ or larger in each set where $g$ is the smallest gain such that at least mob vertices have a gain greater than or equal to $g$.

We select mob vertices at random from this set.

We evaluated the performance of the Mob hypercube
and grid embedding algorithms by conducting an extensive series of experiments on the Connection Machine CM-2. The CM-2 is a massively parallel SIMD (Single-Instruction-Multiple-Data) machine: each processor executes the same instruction at the same time unless it has been disabled. The SIMD approach simplifies debugging, permits an elegant programming style, and does not limit the expressiveness of algorithms. The up to 64K memories on the CM-2 also permits communication along hypercube axes, which is substantially faster than the general router.

Our experimental data is summarized here but reported elsewhere [8]. In our experiments, 1-to-1 mappings of graph vertices to network nodes were used to model VLSI placement problems and standard embedding problems. We also wanted to model bulk-parallelism, as described by Valiant [10], where n virtual processors are embedded into a computing network of size n/log n, and chose 16-to-1 mappings as a rough approximation of log n-to-1 mappings.

We conducted experiments on randomly generated graphs of small (d = 3...16) degrees with up to 500K vertices and 1M edges! The number of iterations needed by the Mob heuristic to reach a fixed percentage above the best-ever embedding cost appears to be approximately constant, on the order of 4000 to 8000, and therefore independent of graph or network size. This holds for hypercube and grid embeddings, and for both 1-to-1 and 16-to-1 mappings. The rate of convergence is shown in Figure 2 for 16-to-1 mappings of the Mob grid and hypercube-embedding algorithms. A good solution is produced rapidly; further improvements can be obtained if enough computation time is available. The number of iterations to achieve a given percentage decreases as the degree increases. The empirical parallel complexity of the Mob heuristic on the CM-2 (on random graphs) is time O(log |E|) with |E| processors.

The absolute speed at which an embedding is produced shows that Mob can be implemented very efficiently on a SIMD-style machine. It takes approximately 36 minutes to find an embedding of a 500K-vertex, 1M-edge graph, the largest graph that would fit into the CM-2, into a 1024 x 512 grid. Due to excessive run times, previous heuristics reported in the literature were able to construct graph embeddings only for graphs that were 100 to 1000 times smaller than those used in our experiments. On small graphs, where simulated annealing and other heuristics have been extensively tested, our heuristic was able to find solutions whose quality was at least as good as simulated annealing.

References

The Effect of Structure in the Mapping Problem
Using Simulated Annealing*

Craig Lee
Lubomir Bic
Department of Information & Computer Science
University of California
Irvine, CA 92717, USA

Abstract

In parallel computation, a common requirement is the mapping of a problem graph of communicating tasks onto a network graph of processing elements such that (1) the communication-distance is minimized, and (2) the problem graph is evenly distributed over the network. When using simulated annealing on this problem, how does the relative structure of the two graphs affect performance? To investigate this question, we used simulated annealing to map four graph types of varying "structure", representing possible problem graphs, onto a graph with a very regular structure, representing a network of processing elements. These results show that higher regularity produces (1) a higher but narrower range of final energy values, (2) a higher distance-energy correlation and (3) a higher degree of ultrametricity.

1 Introduction

Simulated Annealing is a computational technique of finding a near-optimal "solution" to an optimization problem by making random changes in the solution and probabilistically accepting the change depending on whether it improves or degrades the solution [3, 1]. How well simulated annealing performs is dependent on the energy landscape of the configuration space. How can we determine the character of the landscape for a given configuration space? Solla et al. hypothesize that configuration spaces exhibiting a high degree of ultrametricity are more suitable for annealing [7]. Can we use ultrametricity and the correlation between energy and distance to determine if annealing will perform well on a given class of optimization problems? Concomitantly, how does the "structure" of a class of problems affect the degree of ultrametricity and performance? We will investigate these questions by mapping four different graph types that represent problem graphs with a range of structure onto a regular network graph.

2 Four Graph Types

The energy metric we use here for annealing is the distance-variance function as defined in [1]. This attempts to spread a problem graph evenly over the network graph while minimizing communication distance by weighting the two metrics of communication distance and the variance of the number of problem graph edges that can be said to be incident on a network graph vertex as a result of mapping. Ultrametric correlation coefficients are computed as described in [7]. We now investigate: the effect of problem structure on the behavior of simulated annealing and quenching by mapping members of four different graph types, (1) random, (2) geometric, (3) torus, and (4) n-cube, onto a torus. These graph types were chosen to represent a range of "structure". The random graph can be said to have no structure. The geometric graph, as defined below, has more structure with similarities to a torus. The torus and n-cube are very regular in their structure but are similar and dissimilar, respectively, to the network graph.

These graphs are defined in the following way. All problem graphs have 32 vertices and an average vertex degree of 5. This ensures that any difference in numerical energy results from differences in graph structure, i.e., how well the graph can be mapped, rather than just the number of vertices and edges involved. A random graph is generated by defining n vertices and connecting any two with probability p which results in an average degree of (n - 1)p [2]. Given the desired degree of 5, it is easy to compute the required p. A geometric graph is generated by randomly distributing n vertices over the unit square and connecting any two vertices that are contained within a square with side 0 < k < 1 where this smaller square can wrap-around to the other sides of the unit square. This definition is intended to produce a topology similar to a torus. (If wrap-around was not allowed, the generated graph would be more similar to the layout of a printed circuit board as defined in [2].) Here the average degree is 4nk^2. Given the desired degree of 5, it is easy to compute the required k. The problem torus will be a 4 x 8 grid but with extra edges added in a symmetric manner such that each vertex has a degree of 5. Finally, the n-
cube with 32 vertices naturally has a degree of 5. Finally, the network graph will be a $3 \times 3$ torus, each vertex will have a degree of 4 in the typical grid fashion. This network graph may seem small but it is not a degenerate torus and it was deemed necessary to hold down the cost of checking permutations. All of these graphs may seem small but each configuration space has a size of $3^{32}$.

For each problem graph, we will do 100 slower annealings and 100 faster annealings (called quenchings) onto the network graph. In all cases, the same initial mapping will be used but with a different seed for the random number generator. In all cases, an even 0.5/0.5 weighting will be used in the distance-variance energy function. The temperature control parameter has an initial value of 100 ($\approx 50 \times$ the initial energy value) and an exponential decay factor of 0.9. At each temperature, thermal equilibrium will be determined as in [3]. For a problem of size $n$, equilibrium is reached after an average of $n^2$ changes have been attempted or $bn$ changes have been accepted (accepted or not), whichever comes first. The user is free to choose $a$ and $b$ under the constraint that $0 < a \leq b$. For quenching, $a/b = 1/2$. For annealing, $a/b = 128/256$. The algorithm terminates when the configurations at the last three consecutive temperatures show no improvement, i.e., the configuration is "frozen" [5].

For each quenched solution within each graph type, the map permutation with the minimum distance to any annealing solution is found. We then plot the distance versus the energy as a histogram surface as shown in Figures 1 to 4. All annealed solutions are set at distance zero and the distance axis shows how far away the closest quenched solution permutations are. Since the problem graphs all have 32 vertices, all distances are in the range $(0, 32)$. The energy axis, however, is scaled to the minimum and maximum values of all annealings and quenchings and then quantized into 32 bins. The vertical axis then shows the number of solutions in a particular bin. The vertical scale is arbitrary but uniform across all graphs. This graphical display shows at a glance the distribution of the annealed and quenched solutions relative to one another. For comparison, the general distance distribution of all possible configurations from any one configuration is shown as a dotted curve.

Figures 1 to 4 show the following. The random and geometric graphs have very similar results, there is a significant range of annealed energies and an even broader range of energies for the quenched solutions. The additional "structure" in the geometric graph did not significantly change the distribution of solutions found. All of the quenched solutions are in a narrow range of distances from their closest annealed solutions. There is no apparent correlation between the distance between solutions and their difference in energy. In fact, the distribution of quenched solutions seems strongly influenced by the probability of finding a close configuration within the general distribution of configurations.

The torus and $n$-cube graphs, however, tell a different
higher energy can't find a closer annealed solution implies that the configuration space itself has a regular shape, that "valleys" with the same shape have multiple occurrences such that a quenching that lands with energy \( E \) in one valley may be distant from an annealing with energy \( E - \epsilon \) in another valley but has a closer isomorphic permutation in that same valley. Such an interpretation is consistent with the argument given by Solla et al. in support of the correlation between move distance and ultrametricity. However, it is an open question whether these configuration spaces have 'self-similar' or 'fractal-like' valleys, as described in [6], such that similarly-shaped valleys occur with a recursively decreasing size. The observations discussed so far lead us to conjecture that regularity in any optimization problem produces a configuration space that reflects the same structure in which low-lying minima are ultrametrically distributed. This implies that when higher regularity exists, a shorter annealing schedule can be used with a higher probability that the resulting maps will not be far apart in energy.

3 Summary and Conclusions

These experiments have shown that higher regularity in the problem graph produces (1) a narrower but higher range of final energy values, (2) a correlation between move distance and energy, and (3) a higher ultrametric correlation. The fact that quenched solutions with a higher energy can't find a closer annealed solution implies that the configuration space itself has a regular shape, that "valleys" with the same shape have multiple occurrences such that a quenching that lands with energy \( E \) in one valley may be distant from an annealing with energy \( E - \epsilon \) in another valley but has a closer isomorphic permutation in that same valley. Such an interpretation is consistent with the argument given by Solla et al. in support of the correlation between move distance and ultrametricity. However, it is an open question whether these configuration spaces have 'self-similar' or 'fractal-like' valleys, as described in [6], such that similarly-shaped valleys occur with a recursively decreasing size. The observations discussed so far lead us to conjecture that regularity in any optimization problem produces a configuration space that reflects the same structure in which low-lying minima are ultrametrically distributed. This implies that when higher regularity exists, a shorter annealing schedule can be used with a higher probability that the resulting maps will not be far apart in energy.

References


PROPERTIES OF SIMULATED ANNEALING
WITH INACCURATE COST FUNCTIONS

Daniel R. Greening*

ABSTRACT

Inaccurate cost or energy functions appear in many parallel simulated annealing implementations. Such errors alter the equilibrium cost, change the speed to approach equilibrium, and affect the quality of the final result.

We show that identically distributed Gaussian errors do not affect equilibrium. Non-iid Gaussian errors and bounded errors can alter equilibrium; such changes vary with temperature and error magnitude. The convergence speed under bounded errors also depends on temperature and error magnitude.

Finally, we show that constraining errors to a constant factor of the temperature guarantees convergence in a restricted case.

1 INTRODUCTION

Simulated annealing is an algorithm to find nearly-optimal solutions to NP-hard problems [1]. Figure 1 outlines the approach: \( s_0 \) is the initial state, \( G_{s,s'} \) is the generation probability, \( s \) randomly returns an uniformly-distributed random number from \([0,1]\), \( C \) is the cost function, and \( T \) is the temperature. The sequence of temperatures \( T_t \) is the temperature schedule. It is generally presumed that an optimum schedule keeps the average observed cost close to its equilibrium value (not necessarily true, see [2]).

1. \( T \leftarrow T_0 \);
2. \( s \leftarrow s_0 \);
3. while not done
4. \( s' \leftarrow \text{generate}(s) \) with probability \( G_{s,s'} \);
5. \( \Delta \leftarrow C(s) - C(s') \);
6. if \( \Delta < 0 \) \( \lor \) \( \text{random} < e^{-\Delta/T} \) then \( s \leftarrow s' \);
7. \( T \leftarrow \text{reduce-temperature}(T) \);
8. end while;

Figure 1. Simulated Annealing Algorithm

Many parallel implementations involve approximate cost functions [3]. Some sequential implementations also use approximate functions: examples include congestion and wire-length estimates in circuit placement [1].

1.1 Annealing Properties

Define acceptance matrix at time \( t \), \( A(t) \), \( A(s,s') = e^{\min[0,(C(s)-C(s'))/T(t)]} \). Define \( G_{s,s'} \) an inhomogeneous Markov chain \( P \) for simulated annealing,

\[
P(s,s') = \begin{cases} 
G(s,s') & \text{if } s \neq s' \\
1 - \sum_{s \neq s'} G(s,s') & \text{otherwise}
\end{cases}
\]

If we fix the temperature, \( A(t) = A(T) P(t) = P \), and the chain is homogeneous. If it is also ergodic, each state \( s \) has an equilibrium probability \( \rho(s) \) independent of the initial state.

Existing programs attempt to bring state probabilities close to equilibrium at each temperature [1]. Thus, we can gain information about annealing behavior by looking at the homogeneous chain.

We can guarantee ergodicity if these properties hold.

- probability \( \forall s \in S, \sum_{s' \in S} G(s,s') = 1 \) (1)
- coverage \( \forall s, s' \in S, \exists k \geq 1, (G^k)(s,s') \neq 0 \) (2)
- aperiodicity \( \exists s \in S, (G^k)(s,s') \neq 0 \) (3)
- finiteness \( |S| < \infty \) (4)
- symmetry \( \forall (s,s') \in S \times S, (G^k)(s,s') = (G^k)(s',s) \) (5)

(1) states that \( G_{s,s'} \) is a probability vector. (2) guarantees that the annealing chain is irreducible. (3) and (4) make the chain aperiodic and finite. Irreducible, aperiodic, finite Markov chains are ergodic.

1.2 Inaccurate Cost Functions

Parallel algorithms can reduce communication costs by using some stale information, instead of maintaining elaborate synchronization to keep local information up-to-date. Stale data cause inaccuracies in the cost. Experiments show that allowing inaccuracies can improve speed, but can degrade results [4, 5].

Two questions arise: How does the result quality of asynchronous simulated annealing compare to that of sequential simulated annealing? How fast does asynchronous simulated annealing converge to a solution, compared to sequential simulated annealing?

Using a thermodynamic analogy, Grover showed the effect of bounded errors on the partition function [6]. Durand and White analyzed equilibrium properties for bounded errors on a restricted algorithmic class [7]. Gelfand and Mitter showed that state-independent noise, under some conditions, will not affect asymptotic convergence [8].

We do not assume state-independence; most applications appear to exhibit strongly state-dependent errors. We expand the scope further by considering both fixed error bounds and Gaussian errors in the general case, and by examining the effect of bounded errors on speed.

Experiments have hinted that when errors are proportionally constrained to temperature, results improve. Invoking these observations, researchers have modified asynchronous algorithms to obtain better final states [9, 10]. Our last result validates their assumptions.

Proofs for equilibrium properties and convergence speed are omitted (see [11]). Proof of the final result, on convergence to global optima, is retained.

2 EQUILIBRIUM PROPERTIES

We refer to the true cost, \( C(s) \), of state \( s \), and the erroneous cost \( G(s) = C(s) + \epsilon(s) \), of state \( s \). \( \epsilon \) is a random variable dependent on \( s \), thus \( G(s) \) is a random function.

2.1 Bounded Errors

If cost-function errors are bounded, then we have \( \epsilon \leq \epsilon \leq \zeta \).
Theorem 1 Let \( \rho_s(s, T) \) and \( \rho(s, T) \) be the equilibrium probabilities of state \( s \) at temperature \( T \), with bounded errors and without errors, respectively. Then,

\[
e^{(s - s')/T} \rho(s, T) \leq \rho_s(s, T) \leq e^{(s - s')/T} \rho(s, T)
\]  

Equilibrium behavior is often characterized by its “macroscopic properties.” Any macroscopic property, \( F(T) \), is the expected value of some function, \( f(s) \), over the state space,

\[
F(T) = \sum_{s \in S} f(s) \rho(s, T)
\]

Theorem 2 Let \( F(T) \) and \( F(T) \) be equivalent macroscopic properties for function \( f \), at temperature \( T \), with cost functions \( C(s) \) and \( C(s) \), respectively. Then

\[
e^{(s - s')/T} F(T) \leq F_s(T) \leq e^{(s - s')/T} F(T)
\]

A commonly measured macroscopic property, the average cost, is constrained by Theorem 2.

2.2 Gaussian Errors

Simulated annealing typically operates on structures with discrete cost functions: thus the errors appear as discrete values. However, as we add state variables and as the maximum number of uncorrected interdependent moves increases (through increased parallelism or slower update times), the errors can approach a Gaussian distribution.

In many instances, particularly in parallel applications, the probability distribution of the observed cost function \( C(s) \) is reflected in the Gaussian distribution. Then state space, other than those specified by \( C(s) \) and \( C(s) \), are identically distributed, and the resulting homogeneous function \( C(s, t) \) with time and state dependent errors \( \epsilon(t) \), such that

\[
C(s) \pm \epsilon(t) = C(s) + \epsilon(t)
\]

Thus, it is reasonable to investigate the effect of Gaussian random cost function \( C(s) \), with mean \( E[C(s)] = C(s) \). We will show that when the variances of the state costs do not differ greatly, simulated annealing with inaccurate cost functions exhibiting this behavior appear in work by Durand and White [7].

Thus, it is reasonable to investigate the effect of Gaussian random cost function \( C(s) \), with mean \( E[C(s)] = C(s) \). We will show that when the variances of the state costs do not differ greatly, simulated annealing with inaccurate cost functions exhibiting this behavior appear in work by Durand and White [7].

Let the cost of each state \( s \) be \( C(s) = C(s) \), where \( X \) is an independent random variable. Execute the simulated annealing algorithm in Figure 1 with lines 3 and 6 sampling random variables \( C(s) \) and \( C(s) \) instead of bounded random variables, and with \( T \) fixed. If (1)-(5) are satisfied and \( T > 0 \), the resulting homogeneous Markov chain \( P \) is ergodic, and the equilibrium probabilities are given by

\[
\rho_s(i) = \frac{e^{-C(s)/T} P_s(s)}{\sum_{s \in S} e^{-C(s)/T} P_s(s)}
\]

Corollary 1 If all \( X \) are identically distributed, and the resulting Markov chain is ergodic, then \( \rho_s = \rho \) and \( P_s = P \).

Theorem 3 Let \( C(s) \rightarrow R \) be a cost-function, and let its stationary Boltzmann distribution be \( \rho: S \rightarrow [0, 1] \). Consider a random cost function \( \rho: S \rightarrow R \), where each random variable \( \rho(s) \) is an independent Gaussian distribution with mean \( C(s) \) and variance \( \sigma^2(s) \). Let \( \rho(s) \rightarrow [0, 1] \) give its stationary Boltzmann distribution. Then \( \rho(s) \) can be bounded by

\[
e^{(s - s')/T} \rho(s) \leq \rho_s(s, T) \leq e^{(s - s')/T} \rho(s)
\]

Corollary 2 Macroscopic property \( F(s) \) is bounded by

\[
e^{(s - s')/T} F(T) \leq F_s(T) \leq e^{(s - s')/T} F(T)
\]

3 EQUILIBRIUM MIXING SPEED

If \( P \) is a Markov chain and \( \rho(s, T) \) is the stationary probability of state \( s \), define the conductance of a subset \( V \),

\[
\Phi_Y = \frac{\sum_{s \in V} \rho(s, T) \rho(s, T)}{\sum_{s \in V} \rho(s, T)}
\]

In words, the conductance of a subset \( V \) is the conditional probability that a transition will leave \( V \), given that we start in \( V \).

Let \( \rho(V) = \sum_{s \in V} \rho(s, T) \), and let \( S_{1/2} \) be the set of \( V \) with \( \rho(V) \leq 1/2 \).

Define the global conductance as the minimum conductance over all subsets with stationary probability below 1/2, thus,

\[
\Phi = \min_{V \in S_{1/2}} \Phi_Y
\]

This global conductance provides a good measure for annealing speed. It is related to another speed measure, the dominant eigenvalue [12].

Others have examined the effect of adjusting the move spaces to obtain better annealing speed, using conductance and eigenvalues [13, 14]. Here, we show how the size of the errors constrain the global conductance.

Theorem 4 Consider two different annealing chains \( P \) and \( P \), with state space \( S \) and generation probability \( C \) identical. Annealing chain \( P \) has cost function \( C \), and \( P \) has cost function \( C \). They are related by \( C(s) + \epsilon \leq C(s) + \epsilon \). Let \( \Phi \) and \( \Phi \) be the corresponding global conductances. Then,

\[
e^{(s - s')/T} \Phi \leq \Phi \leq e^{(s - s')/T} \Phi
\]

4 CONVERGENCE TO GLOBAL MINIMA

Simulated annealing can be made to converge monotonically to the optimum result, using an appropriate temperature schedule Define global optima set \( S_0 \) such that \( s \in S_0 \rightarrow C(s) = \min\{C(s')\} s' \in S \)

The most general results presume no particular structure to the state space, other than those specified by (1)-(5). We pay a penalty in time for generality. Under these prepositions, a simulated annealing temperature schedule with \( T(t) = c/t \) will converge to the minimum energy states [15, 16].

Proofs of convergence for these general spaces consider sets of local minima \( R^2 \), and presume the cost function returns an integer. Let \( R^0 \) be the set of all local minima. Roughly, \( R^0 \) is the set of local minima which can be escaped to a lower cost by ascending a change in cost of \( k \) (i.e., the height of the cup containing the local minimum). Thus, if the state set is finite and fully connected, there is some \( d \) such that \( R^0 \subseteq S_0 \) (see [16] for a rigorous definition). Theorem 5 Suppose an annealing chain satisfies (1)-(5) and has cost function \( C(s, t) \) with time and state dependent errors \( \epsilon(t) \), such that

\[
C_i(s) + \epsilon(t) \leq C_i(s, t) \leq C_i(s) + \epsilon(t)
\]

Let \( b_i(t) \leq \epsilon(t) \leq \epsilon(t) \leq b_i(t) \), where \( b_i \) and \( b_i \) are constants. Let the temperature schedule be of the form \( T(t) = c\log t \), where \( d \in Z \). Then the algorithm converges in probability to the set of global minima if \( b_i \leq \epsilon(0) \).

Proof. Suppose the transition matrix \( P \) for some Markovian system at time \( t \) is constrained by (16).

\[
-c d P(t) \leq P(t) \leq c d P(t)
\]

where \( c_1 \) and \( c_2 \) are positive constants. Assume for some integer \( d \geq 0 \) that (17) and (18) hold.

\[
\sum_{t=0}^{\infty} e^{-d/t} < \infty
\]

We can then conclude that (19) and (20) are true [16].

\[
\Phi \leq \Phi \leq \Phi
\]

(20)

Let \( c_1 = e^{b_1} \) and \( c_2 = e^{b_2} \). These values satisfy (15) and (16).

Choose \( d \) so that \( R^0 \subseteq S_0 \). Such a \( d \) must exist, since (1)-(5) are satisfied. Let \( T(t) = c\log t \). This satisfies (17) and (18). By (19) the erroneous simulated annealing algorithm converges in probability to the set of global minima. 

825
5 CONCLUSION

Calculation errors in simulated annealing affect the equilibrium cost and the speed at which equilibrium is reached. Analytic results presented in this manuscript constrain equilibrium properties with a bounded or Gaussian random cost function. In addition, we constrain annealing speed with a bounded cost function.

For bounded errors, the range of the observed cost increases relative to the true cost, equilibrium properties and annealing speed diverge exponentially from true equilibrium and sequential annealing speed. For Gaussian errors, as the difference between the lowest and highest variances increases, equilibrium properties diverge exponentially. Finally, all these quantities diverge exponentially with the inverse-temperature.

Researchers have speculated that tuning errors to a constant factor of temperature helps the system converge. Our final result shows that guess to be correct, at least in the case of inverse logarithmic temperature schedules.

This work was supported by the IBM T.J. Watson Research Center. I gratefully acknowledge the comments of Frederica Darema, John Tsitsiklis, Albert Boulanger, and Griff Bilbro.

REFERENCES

Time-homogeneous Parallel Annealing Algorithm

Kouichi Kimura Kazuo Taki
Institute for New Generation Computer Technology
1-4-28 Mita, Minato-ku, Tokyo 108, Japan

1 Introduction

Simulated annealing is a general and powerful technique to solve difficult combinatorial optimization problems [1]. It consists of many iterative steps. Each modifies the current solution randomly and accepts it with probability \( \min(1, \exp(-\Delta E/T)) \). Here \( \Delta E \) represents the gain obtained by the proposed modification in terms of the energy (objective function) \( E \), and \( T > 0 \) is the temperature which is gradually reduced according to a cooling schedule.

Unfortunately, the theoretically optimal cooling schedule, which guarantees the convergence to the optimal solution, proves to be too slow for practical use [2]. Cooling schedules with geometrically decreasing temperatures are often used in applications. To obtain more elaborate cooling schedules is an active area of research [3].

In this paper we propose a new parallel simulated annealing algorithm. Each processor maintains one solution and performs the annealing process concurrently at a constant temperature that differs from processor to processor, and the solutions obtained by the processors are exchanged occasionally in some probabilistic way. An appropriate cooling schedule is automatically constructed from the set of the temperatures that are assigned to the processors. Thus we can avoid the task of carefully reducing the temperature according to the time, which is essential for the performance of the conventional sequential annealing algorithm.

In this paper we propose a scheme of the probabilistic exchange of solutions and justify it from the viewpoint of probability theory. We have applied our algorithm to a graph-partitioning problem. Results of experiments, and comparison with those of the sequential annealing algorithm and the Kernighan-Lin algorithm, are discussed.

2 An Annealing Algorithm Parallelized in Temperature

2.1 Outline of the Algorithm

The basic idea is to use parallelism for various temperatures, to perform annealing processes concurrently at various temperatures instead of sequentially reducing the temperature according to the time.

The outline of the algorithm is as follows. Each processor maintains one solution and performs the annealing process concurrently at a constant temperature that differs from processor to processor. After every \( k \) annealing steps, every pair of the solutions from the processors with adjacent temperatures are exchanged with some probability \( p \), which is distinct for each pair. The algorithm can be stopped after any large number of steps and we will find a well-optimized solution on the processor that has the lowest temperature. We refer to \( f = 1/k \) as the frequency of (probabilistic) exchanges and \( p \) as the probability of exchange.

Since exchanging the solutions between processors with different temperatures is nothing but changing the temperature for each participant solution, each solution will select its appropriate cooling schedule dynamically through successive competitions with others for lower temperature. However, since the temperature on each processor remains constant, the algorithm itself is time-homogeneous. Thus we can avoid the task of carefully reducing the temperature according to the time, which is essential for the performance of the conventional sequential annealing algorithm. In other words, this algorithm automatically decides how many steps should be taken at each temperature, the majority of steps should be devoted to some critical temperatures.

However, it is necessary to allocate an appropriate temperature to each processor beforehand. Namely, we have to specify a set of temperatures, from which the algorithm will construct a cooling schedule. This set should be chosen wisely according to the estimation of the equivalence (static) relation between the temperature and the energy. It must cover the region of temperature, only in which the equilibrium energy varies virtually. Here the concepts of the scales by S. White will be useful [4].

2.2 Probability of Exchange

Investigating the necessary condition which the probabilistic exchange must satisfy, we determine the probability of exchange.

Imagine that the annealing process is performed independently at each processor at a distinct constant temperature. Then the distribution of the solution in each processor approaches Boltzmann distribution of the respective temperature [3]. The lower the temperature is, the better the solution that will be found, but after a longer time.

Now we introduce probabilistic exchanges of the solutions between the processors and intend to accelerate the convergence of the solutions so that we can find a better solution at the lowest temperature more quickly.

Let \( p(T, E, T', E') \) denote the probability of the exchange between two solutions with energy \( E \) and \( E' \) at temperatures \( T \) and \( T' \). Since we expect a better solution at a lower temperature, we define \( p(T, E, T', E') = 1 \) if \( (T - T')(E - E') < 0 \).

On the other hand, if \( (T - T')(E - E') \geq 0 \), \( p(T, E, T', E') \) is uniquely determined as follows. In order to accelerate the convergence, a probabilistic exchange of the solutions must not disturb the equilibrium distribution. Hence the detailed balance equation must hold between the distributions before and after the exchange:

\[
\frac{1}{Z(T)} \exp\left(\frac{E}{T}\right) \cdot \frac{1}{Z(T')} \exp\left(-\frac{E'}{T'}\right) \cdot p(T, E, T', E') = \frac{1}{Z(T)} \exp\left(-\frac{E'}{T'}\right) \cdot \frac{1}{Z(T')} \exp\left(-\frac{E}{T}\right) \cdot 1
\]

where \( Z(T) \) denotes the partition function. Therefore we obtain

\[
p(T, E, T', E') = \begin{cases} 
1 & \text{if } \Delta T \Delta E < 0 \\
\exp\left(-\Delta T \Delta E \frac{E'}{T'}\right) & \text{otherwise}
\end{cases}
\]

where \( \Delta T = T - T' \), \( \Delta E = E - E' \). Note that \( p(T, E, T', E') > 0 \) for all \( T \neq T' \). This means that a solution can go through a non-monotonic cooling schedule.

This probability is quite different from that of choosing a solution-temperature pair in the systolic statistical cooling algorithm by E. Aarts et al. [5]. The advantage of the former is that it does not contain the partition function and hence can be computed easily.

2.3 Monotonic Convergence Property

We verify that each probabilistic exchange of solutions in fact accelerates the convergence of the algorithm.
Let \( p \) denote the distribution of the solutions at an arbitrary time. It will change into \( pA \) after one annealing step at each processor, or into \( pC \) after one probabilistic exchange for each pair of solutions, where \( A \) and \( C \) are the respective transition probability matrices \([6]\). Let \( \pi \) denote the equilibrium distribution. It can be shown \([6]\) that

\[
D(\pi(p) \| pA) \geq D(\pi(p) \| pC)
\]

where \( D(\pi(p) \| p) \) denotes Kullback-Leibler divergence of \( \pi \) and \( p \), which represents the discrepancy between them \([7]\). Here strict inequalities hold unless \( p = \pi \). Moreover \( D(\pi(p) \| p) \to 0 \) follows from the observation in the subsequent subsection.

Hence the distribution of the solutions monotonically approaches the equilibrium distribution during the execution.

### 2.4 Time-homogeneity

The above algorithm is *time-homogeneous*: it has no control parameter to change over time. This has two implications.

Firstly, the behavior of the algorithm is described in terms of a time-homogeneous Markov chain. In general it is an irreducible and acyclic (ii) How should we assign temperatures to the processors?

Secondly, in executing the algorithm, we can stop it at any time and examine whether a satisfactory solution has already been obtained. If one has not, we can resume it again for a better solution, and can just continue it as long as we like. In contrast, in the conventional simulated annealing it is necessary to re-schedule the temperature when we resume it, once it has entered the lowest temperature.

### 3 Experimental Results

We have implemented our algorithm for a graph-partitioning problem on the Multi-PSI/V2 \([9]\), an MIMD parallel machine with 64 PEs.

**Graph-partitioning problem.** Given a graph \( G = (V,E) \), define a label on the vertices \( V \to \{-1,1\} \) so as to minimize \( E \), where

\[
E = - \sum_{(u,v) \in E} \lambda(u)v + c \cdot (\sum_u \lambda(v))^2, \quad (c > 0: \text{constant})
\]

This is an NP-hard problem. Kernighan-Lin algorithm efficiently gives its approximate solutions \([8]\).

For a random graph \( G \) with 400 vertices and 2004 edges, we compare the results given by our algorithm with those by other methods [Fig.1].

(a) **Time-homogeneous parallel annealing.** All 63 processors performed 20,000 annealing steps each at distinct constant temperatures. The highest and lowest temperatures are determined empirically, and the other temperatures are determined so that adjacent ones have the same ratio. As for frequency of exchanges \( f \), we examined various values ranging from 1/20,000 to 1/2. Each point represents the average over 30 runs with different sequences of random numbers.

(b) **Sequential annealing:** The cooling schedule consists of exactly the same sequence of 63 temperatures as above. 20,000 annealing steps are performed, which are divided equally between the 63 temperatures.

(c) **Simple parallel annealing.** Each of 63 processors executes the sequential annealing algorithm described in (b) using a distinct sequence of random numbers. The result is the best solution obtained by them.

(d) **Kernighan-Lin:** Kernighan-Lin algorithm is repeatedly applied several times until convergence.

We made the following observations from [Fig.1].

1) (a) gives the best solutions for a wide range of the frequency of exchanges: \( 1/1000 \leq f \leq 1/2 \). Hence this algorithm is not sensitive to the value of \( f \) except for values that are too small. However a too large value of \( f \) incurs a high cost in exchanging the solutions between the processors. The execution time for \( f = 1/1000 \) was less than 8% greater than that for \( f = 1/100 \) \([6]\).

2) Since 20,000 annealing steps are relatively small, (b) gives a worse solution than (d). However, in (a), the algorithm probabilistically selects an appropriate cooling schedule with 20,000 steps and gives a better solution.

3) Note that the total number of annealing steps in (a) and that in (c) are the same. (a) outperforms (c) unless \( f \) is too small.

### 4 Conclusion and Future Works

We have proposed the time-homogeneous parallel annealing algorithm, in which an appropriate cooling schedule is automatically and probabilistically constructed from a set of temperatures.

The behavior of this algorithm is theoretically tractable, since it is described in terms of a time-homogeneous Markov chain. In particular we have proved its monotonic convergence property.

We have experimentally observed that this algorithm automatically constructed a better cooling schedule than that which assigned the same number of annealing steps at each temperature. We also observed that this algorithm is robust for the choice of the frequency of exchanges.

### 5 Acknowledgments

We would like to thank N. Ichiyoshi, K. Rokussawa, and E. Sugino for valuable discussions.

### References


**Fig.1. energy vs frequency of exchanges**

(c) Kernighan-Lin

(b) sequential annealing

(a) time-homo, parallel annealing

(d) simple parallel annealing

-1760 -1740 -1720 -1700 -1680 -1660 -1640 -1620 -1600

-1200 -1000 -800 -600 -400 -200 0 200 400 600

-1200 -1000 -800 -600 -400 -200 0 200 400 600

-1200 -1000 -800 -600 -400 -200 0 200 400 600

-1200 -1000 -800 -600 -400 -200 0 200 400 600
Parallelization of the Simulated Annealing Algorithm:
Application to the Placement Problem
Bernard-Virot (LIFO)
University of Orleans BP 6759
45067 Orleans Cedex 2 France

Abstract: We present a method of parallelization of the simulated annealing algorithm, applied to an instance of the chip placement problem. We give a mathematical evaluation for the synchronization cost and for the speedup of the method. We show that, for each stage of the algorithm, there exists an optimal number of processes, which depends only on a small number of measurable parameters. So, in order to obtain the best speedup, our method makes the number of employed processes vary dynamically during the execution of the algorithm.

An implementation of the method on a shared memory architecture is described, as well as its application to a real-size problem: the placement of a graphic card made up of 272 chips and 638 equipotentials.

1. The Metropolis algorithm

Let us recall briefly the classical optimization method based on the Metropolis algorithm [1]. Let $E$ be the energy function to minimize, defined on the state space $\Theta$ of a system. For each state in $\Theta$, we define a set of neighboring states, and we call elementary move each transformation bringing a state $x$ to a neighboring state $y$. We give a transition matrix $Q = (q_{xy})$ on $\Theta \times \Theta$, markovian, symmetric and irreducible, such that $q_{xy} > 0$ if and only if $x$ is a neighbor of $y$ and $x \neq y$.

We model the dynamic system by a Markov chain $(X_n)$ over $\Theta$, controlled by a parameter $T$ called temperature, and defined in the following way [2]. Suppose $X_0, \ldots, X_n$ have already been built. We choose at random an elementary move bringing to a state $Y_n$, according to the probability law

$$\text{Proba}(Y_n = y|X_0, \ldots, X_n) = q_{X_n y}.$$  

We impose, with probability 1,

$$(X_{n+1} = Y_n \text{ or } X_{n+1} = X_n),$$  

the choice being random according to the law

$$\text{Proba}(X_{n+1} = Y_n|X_0, \ldots, X_n, Y_n) = \min\left(1, \exp\left(-\frac{\Delta E}{T}\right)\right).$$

Here $\Delta E$ denotes the energy variation corresponding to the elementary move bringing from $X_n$ to $Y_n$. This move will be called a trial move. If $X_{n+1} = Y_n$, then we say it is an acceptable move. Otherwise ($X_{n+1} = X_n$), it is a rejected move.

Now, if we take for $T$ a decreasing function of $n$ tending to 0 sufficiently slowly, ($T = T_n \geq \frac{C}{n}$, where $C$ is a suitably chosen constant), then we can prove that the law of $X_n$ converges to a measure supported by the set of the absolute minima of the energy function $E$ (cf. [2]).

2. Parallelization of the placement problem

In the sequel we study the parallelization of the Metropolis algorithm, applied to a specific type of problem, namely the chip placement problem. One can think of two dual parallelization methods:

- **Data partitioning:** One distributes the data (chip positions) among the processes.
- **Tasks partitioning:** One distributes the tasks among the processes, the data being shared.

With the data partitioning method a process can move only its own chips. In order to accept or reject a chip move, according to the Metropolis algorithm, one has to know the current length of all the equipotentials connected to that chip. But two chips belonging to two different processes can belong to the same equipotential. So, if erroneous computations of the energy variation are not admitted, then one must forbid simultaneous moves of such chips. Even for a small number of processes the synchronization time becomes prohibitive. For example, suppose that each equipotential connects 4 chips on the average, and each chip belongs to 10 equipotentials. Then, for every trial move of a chip, a process must lock those of the $4 \times 10 = 40$ chips which belong to other processes, i.e. 15% of the total number in our case. If more than seven processes run simultaneously, then there exists an undesirable waiting time due to the lack of available chips. Let us point out that this method leads to a dynamic partitioning problem which can, in turn, be solved by simulated annealing [3].

With the tasks partitioning method the chip positions are shared data. One has to decide what exactly we mean by a task. It may mean completing a move (coarse grain decomposition), or finding out an acceptable move (medium grain decomposition). One can even think of a fine grain decomposition where two or more processes cooperate for the computation of the energy variation involved by a single trial move [4]. The coarse grain decomposition clearly requires a mutual exclusion protocol in order to avoid contradictory decisions. Moreover, while a chip is moved, all the chips belonging to the same equipotentials must be locked. The fine grain decomposition is "processor consuming" since each trial move involves the cooperation of several processes. It is suitable for a massively parallel architecture, where a processor can be efficient only for simple actions. The medium grain decomposition involves no lock at all, and only minimal synchronization. The parallelization method we studied corresponds to this choice. In the sequel, we will call it the parallel trials method.

3. The parallel trials method

The method was previously used by S. A. Kravitz, R. A. Rutenbar [5], and also by E. Aarts and J. Korst. Starting from a common initial state, the processes build independent Markov chains, until one of them (at least) obtains a number $S$, fixed in advance, of successive acceptable moves according to the Metropolis algorithm. Such a process will be called a winner. Then, all the processes synchronize. One chooses at random exactly one of the winners and lets its current configuration (resulting from its $S$ acceptable moves) be communicated to the others processes. Then, one starts again from the new common state. Thus, each process executes the following algorithm.
The variable my_count is private, whereas found is shared. Termination_test is the test defining the termination of the algorithm. In order to avoid deadlock, it must have the same value in all the processes. Test(m) is a function returning TRUE if the move m is acceptable and FALSE otherwise. In the procedure Synchronize the processes synchronize and agree on the starting configuration that they will use in the next step.

The role of the parameter S is crucial, because there are different phases in the execution of the algorithm. At low temperature, few of the trial moves are acceptable. So, synchronizing the processes for each acceptable move (S = 1) involves only a small waste of time, with respect to the synchronization time involved by a greater value of S. Moreover, choosing $S = 1$ maximizes the speedup produced by independent parallel searches of an acceptable move.

At high temperature the situation is quite different. Most of the moves are acceptable. So, the speedup produced by the parallel searches of an acceptable move is not very significant. If we synchronize the processes as soon as an acceptable move is found, then the synchronization time can become prohibitive. This will moreover favor the moves for which the decision is the fastest (for example, the translation of a chip with a small number of connections). In the opposite, if we choose $S$ sufficiently large, then we obtain Markov chains long enough so that the total computation time of a chain can be considered independent of the choice of the trial moves. To sum up, at high temperature, the parameter $S$ plays the following roles:

- reduce the synchronization overhead: the greater is $S$, the smaller the number of process synchronization;
- make the acceptance probability independent of the types of the moves: for $S$ sufficiently large, the computation time to find out a number $S$ of acceptable moves can be considered independent of the complexities of the tried moves.

4. A small model

4.1 Local analysis (constant temperature)

In this section, we fix a low temperature $T$ and we take $S = 1$. We denote by $\alpha$ the mean acceptance rate, that is the probability that a trial move be acceptable, following the Metropolis algorithm. If $T$ is given, then $\alpha$ is constant. Let us call $t_N$ the mean waiting time for the first acceptable move, when $N$ processes make parallel independent trials. If $\alpha$ is sufficiently small, then we may estimate that

$$t_N = \frac{1}{1 - (1 - \frac{1}{2})^N} \approx \frac{t_1}{N}$$

Let us denote by $s_N$ the mean time needed by $N$ processes to synchronize and to agree on a new starting configuration. We denote by

$$r_N \approx \frac{t_1}{N} + s_N$$

the total mean time needed to achieve a move.

We assume $s_N$ is proportional to the number of processes:

$$s_N = aN$$

(we postpone the discussion of the validity of the hypothesis $s_N \approx aN$ until the end of the section 4), whence the formula

$$r_N \approx \frac{t_1}{N} + aN$$

(1)

Note that, since $t_1$ is the mean waiting time for an acceptable move with exactly one process, we have

$$t_1 = \frac{c}{\alpha}$$

(2)

where $c$ is a constant, which can be viewed as the sum of the times to choose at random a move, to compute the corresponding variation of energy, and to take the decision of acceptance or rejection. The parameter $c$ depends on the implementation of the algorithm, on the architecture, and on the instance of the placement problem.

An elementary computation shows that $r_N$, considered as a function of $N$, reaches its minimum $r_{N_{\text{opt}}}$ for

$$N = N_{\text{opt}} \approx \sqrt{\frac{t_1}{a}} = \sqrt{\frac{c}{\alpha}}$$

(3)

and that we have

$$r_{N_{\text{opt}}} \approx 2\sqrt{at_1}$$

(4)

Therefore, for a fixed temperature $T$, there exists an optimal number of processes $N = N_{\text{opt}}$. Moreover, $N_{\text{opt}}$ is approximately proportional to $\alpha^{-\frac{1}{2}}$.

For this optimal value of the number of processes, the speedup $G$ we obtain can be defined and computed in the following way:

$$G = \frac{r_1}{r_{N_{\text{opt}}}} \approx \frac{1}{2} \left( \frac{t_1}{a} + \sqrt{\frac{a}{t_1}} \right)$$

(5)

When the temperature is sufficiently low, the acceptance rate $\alpha$ is small and, therefore, $t_1$ is large. Thus, we may write

$$G \approx \frac{1}{2} \sqrt{\frac{t_1}{a}} = \frac{1}{2} N_{\text{opt}}$$

(6)

Equation (3) above shows that $G$ is then proportional to $\alpha^{-\frac{1}{2}}$.

4.2 Global analysis (variable temperature)

We have chosen to decrease the temperature by steps, following the exponential law $T_k = (0.90)^k T_0$ where $T_k$ denotes the $k$th step's temperature.

In order to determine the length of the $k$th step, one may consider two types of rules. Let $L_k$ denote the length of the Markov chain and $L_1$ the number of accepted moves, both
counted from the beginning of the current step, and let $C_0$ and $C_1$ be suitably chosen constants (initially, $L_0 = L_1 = 0$, and we always have $L_0 \geq L_1$).

rule 1: The $k^{th}$ step ends when $L_0 \geq C_0$ or $L_1 \geq C_1$.

rule 2: The $k^{th}$ step ends when $L_1 \geq C_1$.

Let us remark that, in order to compute $L_0$, one has to decide how to take into account the moves rejected by the parallel processes. An extensive discussion of this problem can be found in [6].

At low temperature, under the first rule, the steps are cut down into a constant length determined by $C_0$, leading to an exponential decrease of the temperature. For our instance of the placement problem the experiments done with this rule highlighted the trap of the annealing in a local minimum. The second rule makes the steps longer, proportional to $tNk$, where $\alpha$ denotes the current acceptance rate, slowing down the temperature decrease. Our experiments done with this second rule led to a mean energy 20% smaller than with the first rule. In the sequel we therefore consider only the second rule. The need for a very slow temperature decrease seems mainly due to the constraints involved in our problem. We postpone the discussion of this point until section 5.

Let us denote by $\alpha_k$ the mean acceptance rate in step $k$, by $tN_k$ the corresponding mean waiting time for the first acceptable move with $N$ processes in parallel, and by $rN_k$ the mean time needed to achieve a move. Equation (5) above shows that the optimal speedup $G_k$ for the $k^{th}$ step is then given by the relation

$$G_k \approx \frac{1}{2} \left( \sqrt{\frac{tN_k}{\alpha}} + \sqrt{\frac{1}{\alpha}} \right)$$

or, since $tN_k = \frac{c}{\alpha_k}$,

$$G_k \approx \frac{1}{\alpha_k} \left( 1 + \frac{a}{c} \right)$$

Therefore, if we choose dynamically, for every step, the optimal number $N_{opt}$ of processes given by equation (9), then we have

$$rN_k = \frac{1}{G_k} r_{1,k} = \frac{1}{G_k} \left( \frac{c}{\alpha_k} + a \right) ,$$

whence, finally the formula

$$rN_k \approx 2 \sqrt{\frac{ac}{\alpha_k}}$$

Since all the steps contain the same number of accepted moves we may define $G_{glob}$ for $K$ successive steps by the relation

$$G_{glob} = \frac{\sum_{k=1}^{K} r_{1,k}}{\sum_{k=1}^{K} rN_k}$$

Using the equation (7) above, we can write $G_{glob}$ as a function of the sequence of acceptance rates $\alpha_k$, $1 \leq k \leq K$.

$$G_{glob} \approx \frac{1}{2} \sqrt{\frac{E}{a}} \left( \sum_{k=1}^{K} \frac{1}{\sqrt{\alpha_k}} \right) \left( \sum_{k=1}^{K} \frac{1}{\sqrt{\alpha_k}} \right)^{-1}$$

Equation (8) allows the effective compuation of $G_{glob}$, since $a$, $c$ and $\alpha_k$ are easily measurable.

4.3 Discussion

In this section we discuss the relations of our method with the architecture and the implementation.

The algorithm contains a random choice among the winners, after every $S$ accepted moves. On a shared memory architecture this choice can be made simply by a specialized process (master). On a distributed architecture it would probably be better to use a more symmetric algorithm, the winners electing one of them by means of an election algorithm.

We made the assumption that the synchronization and updating mean time $s_N$ is proportional to the number of processes. Let us remark that $s_N$ depends on the implementation of the algorithm and on the architecture, but it does not depend on the instance of the placement problem. On a shared memory architecture, if the synchronization protocol uses a critical section, then $s_N$ is bounded below by a constant multiplied by the number $N$ of processes. So, our hypothesis corresponds to a good implementation of a protocol belonging to this class. On a distributed architecture the situation can be quite different. For example, on a grid of $N$ processors, the time to broadcast a value, and thus to update the data structures, is proportional to $\sqrt{N}$, and no critical section is needed. However, the corresponding proportionality constant is likely to be much greater that of the shared memory case.

5. The placement problem

The previous algorithm is implemented on a Sequen- machine (Balance 8000). It is a 32 bits shared memory multiprocessor architecture, with a single bus and a cache memory. The machine we used had eight processors. We studied one instance of the chip placement problem, namely the placement of a graphic card made of 272 chips and 638 equipotentials. The energy function we had to minimize has the following form

$$E = E_f + pE_r + qE_d$$

where we denote by

$E_f$ the total length of the wires,

$E_r$ the sum of the overlapping areas of the chips taken two by two,

$E_d$ the sum of the areas of the chips parts extending beyond the limits of the board,

$p$ and $q$ two adjustable weights.

We have chosen the following elementary moves: the exchange of two chips, the rotations of angle $\frac{\pi}{2}$, the translations small enough to ensure that a chip is never entirely outside the board. One imposes several constrains:

- Some chips, such as the connectors, must remain fixed.
- There are forbidden areas, that no chip may intersect.
- When the annealing ends, we want to get $E_r = E_d = 0$; all the chips must fit in the rectangle formed by the board, without any overlap.

In order to satisfy (asymptotically) the latest constraint $p$ and $q$ must tend to $\infty$ with the length $n$ of the Markov chain, from the beginning of the simulated annealing. Let us denote
by $p_n$, $q_n$ and $T_n$ the current values of the weights $p$, $q$, and of the temperature $T$. D. Geman and S. Geman [7] showed that the ratios $\frac{p_n}{T_n}$ and $\frac{q_n}{T_n}$ must tend to $+\infty$ very slowly, namely

$$\frac{p_n}{T_n} \leq C \log n \quad \text{and} \quad \frac{q_n}{T_n} \leq C \log n,$$

where $C$ is a suitable constant. In particular, both the decrease of $T_n$ and the growth of the weights $p_n$ and $q_n$ must be very slow. If we take $T_n = \frac{C}{\log n}$, then the above constraint on $\frac{p_n}{T_n}$ would imply $p_n \leq C^2$ and therefore, the condition $p_n \to +\infty$ is not satisfied. This explains why temperature steps with constant length are not suitable for optimization problems with constraints (see section 4.2 above).

6. Numerical values

6.1 Local analysis

We determined experimentally the parameters we introduced in the section 4, by computing a mean on 5000 iterations at a constant temperature.

6.1.1 Synchronization and updating mean time $s_N$

![Graph showing synchronization and updating mean time](image)

Fig. 1 The theoretical graph is drawn with surrounded squares.

We deduce from this graph (Fig. 1) the estimated value for $a$:

$$a \approx 0.7$$

The divergence of the theoretical graph from the experimental one for $N \geq 7$ is due to the processors' saturation. In fact, the hypothesis $s_N = aN$ is legitimate only if each process gets a processor as soon as it is ready to perform a step. This is not true when the number of processes is close to the number of existing processors (eight), because in the UNIX system, at least one of the processors has to run the kernel's processes.

0.1.2 Move achievement mean time $r_N$

- $\alpha = 0.45$, $S = 1$. The mean acceptance rate is $\alpha = 0.45$, corresponding to a relatively high temperature, and we synchronize the processes as soon as an acceptable move is found.

![Graph showing move achievement mean time](image)

Fig. 2 $\alpha = 0.45$, $S = 1$

We can thus see that choosing $S = 1$ at high temperature slows down the algorithm (Fig. 2).

- $\alpha = 0.45$, $S = 4$. The mean acceptance rate is $\alpha = 0.45$, and we synchronize when a process has found 4 successive acceptable moves.

![Graph showing move achievement mean time](image)

Fig. 3 $\alpha = 0.45$, $S = 4$

The local speedup is optimal when $N = 4$. Its measured value is $G = \frac{r_4}{r_1} = \frac{3.5}{2.78} \approx 1.2$ (Fig. 3).
• $\alpha = 0.05$, $S = 1$. The mean acceptance rate is $\alpha = 0.05$, corresponding to a low temperature, and we synchronize as soon as a process has found an acceptable move.

6.2.1 Global speedup

The graph (Fig. 5) above shows the measured variations of the mean acceptance rate considered as a function of the temperature. Using its values and the relation (8), one can compute the global optimal speedup $G_{glob}$, obtained when the algorithm runs, for each temperature, the optimal number of processes. For the instance of the placement problem we studied one obtains the value

$$G_{glob} \approx 7.5$$

The graph (Fig. 5) shows the estimated optimal number of processes, computed by the relation (3). Recall that the machine we used had 8 processors. Thus, one can see that we could not run this optimal number of processes for temperatures lower than 30.

6.2.2 Energy

We performed five simulated annealing for the same ins-
tance of the placement problem. We obtained a mean final energy $m = 41177$ with a standard deviation $\sigma = 1011$. Let us remark that the initial placement of the card was done "by hand" in about a week, resulting in a final energy $30\%$ higher.

The following graph (Fig. 7) gives a typical example of the variation of the mean energy versus the temperature.

![Graph showing energy vs. temperature](image)

**Fig. 7** energy

0.3 Feasibility

The card's description file, in a standard format, is read by the program through a parser. After the placement is completed, the program produces a file in the same format, describing the final card. This last file is then processed automatically by a routing software, demonstrating the feasibility of our placement solution.

Conclusion

The results above show that the parallelization method we studied is not a massive parallelization method. But, it gives a significant speedup at low temperature.

In order to obtain the best speedup, one must adapt dynamically the number of processes to the current mean acceptance rate. The notion of adaptive strategy was introduced by S. A. Kravitz and R. A. Rutenbar. In [5] they juxtapose different parallelization methods, leading to complex and, a priori, not optimal schedules. Our theoretical model allows us to compute the optimal number of processes, and the speedup, for each step of the algorithm.

The algorithm and the theoretical model are general enough to handle both shared memory and distributed architectures. They were tested only on a shared memory computer. However, on a distributed architecture, the synchronization and updating mean time, as a function of the number of processes, is likely to behave differently, leading to new estimates for the optimal number of processes and for the speedup. We plan to experiment on a distributed computer based on Transputers.

Acknowledgements.

It is a pleasure to acknowledge the help of Jean Jacques Levy at I.N.R.I.A., who put at our disposal a Sequent machine. We also thank Philippe Matherat for the file describing his graphic card, and Marcin Skubiczewski who wrote the parser reading the file.

The present work was partially supported by the French GRECO C3.

**BIBLIOGRAPHY**

THREE-DIMENSIONAL ELECTROMAGNETIC PARTICLE-IN-CELL SIMULATIONS
OF PHYSICAL DEVICES

ALAN MANKOFSKY
Applied Physics Operation
Science Applications International Corporation
1710 Goodridge Drive
McLean, Virginia 22012 USA

Abstract - We present an overview of three-dimensional electromagnetic particle-in-cell (PIC) simulation techniques for vector supercomputers, and their application to the realistic design of physical devices. We first describe the fundamental building blocks of electromagnetic PIC codes. We then discuss code architectures for combining these components into a working design tool, using the ARGUS system of codes as a specific case study. Finally, applications and examples are discussed.

I. INTRODUCTION

For more than twenty years, computers have been used to simulate systems of charged particles subject to both applied electromagnetic fields and to self-consistent fields generated by the particles themselves. Codes of increasing generality and complexity have evolved over this period. Today, with the widespread availability of vector supercomputers, three-dimensional codes which can self-consistently treat relativistic particles, electromagnetic fields, complex multimaterial structures embedded on the computational mesh, and realistic boundary conditions have become essential tools in the physics and engineering communities. The generality of such codes permits state-of-the-art modeling of, for example, microwave tubes (klystrons, magnetrons, traveling-wave tubes), accelerators, electron guns and electron optics systems, solid-state devices, and antennas. In effect, supercomputers have transformed electromagnetic PIC codes from research projects into truly cost-effective design and problem solving tools.

II. PHYSICS MODELS

The basic building blocks of an electromagnetic PIC code include the following:

Field solvers - These consist of both direct and iterative algorithms for solving particular subsets of Maxwell's equations. The full electromagnetic set and the electrostatic limit are the most common, with the magnetostatic and magnetoinductive approximations useful under certain circumstances. In addition, the full set of Maxwell's equations can be solved either in the time domain (as an initial value problem) or in the frequency domain (as an eigenvalue problem). The latter technique is an efficient and accurate way of determining electromagnetic spectra.

Particle pushers - These perform temporal orbit integration of relativistic particle species, and accumulate the source terms needed by the field solver. The particle equations of motion may include elastic and inelastic scattering from background species, interactions with matter, and terms describing rate processes.

"Kitchen sink" physics - Algorithms for describing surface physics, radiation, and other "dirty" processes are developed as needed. These can include phenomenological descriptions as well as more fundamental models.

Geometry specification and visualization - While not physics modules in the strict sense, these components are critical to a working three-dimensional PIC model. Complex geometries can be extremely difficult to set up in three dimensions, similarly, extracting some essential physical behavior form complicated three-dimensional field topologies and particle flows is often a daunting task. Combinatorial geometry tools (allowing for basic Boolean operations to be performed on three-dimensional objects) and advanced visualization packages (often workstation-based) are essential parts of an overall design system.

III. CODE ARCHITECTURE

Three-dimensional electromagnetic PIC models must handle vast quantities of data: a typical moderately-resolved simulation may require of the order of 100 million words of storage. Domain decomposition algorithms must be coupled with memory management and data management techniques for optimizing the use of fast memory for each calculation and for efficiently moving data between memory and disk as the calculation proceeds.

In SAIC's ARGUS code, for example, problems are decomposed into spatial regions known as field blocks, which are then arranged on a set of generalized lattices known as logical supergrids. The size of each field block is chosen so that the block will fit easily into memory; data-handling routines move blocks between disk and memory as needed. Disk I/O is overlapped with computations wherever possible. Algorithms used by both the field and particle routines permit global solutions in the entire physical domain by performing sequence-independent operations in each field block, followed by sharing of data at interfaces. Solutions in individual blocks may be advanced either synchronously or asynchronously; this architecture is naturally suited to parallel processing.

IV. APPLICATIONS AND EXAMPLES

Three-dimensional electromagnetic PIC simulations are now widely used for research and design in a number of diverse areas. We briefly describe two different types of calculations that have been performed with the ARGUS code.

Microwave device and antenna design - The process of measuring the rf properties and mode structure of a microwave configuration with no applied voltages (i.e., with no particles in
it) is known as cold testing. It is typically the first series of measurements made when developing a new device. Once the cold test properties of the structure are satisfactory, the designer will proceed to hot testing, or measurements where the device is under voltage and particles are present. As described above, cold testing can be performed in either the time domain or the frequency domain.

ARGUS has been used to predict the cold test behavior of numerous devices, including microwave cavities, high-power rf tubes, and high-frequency component packages. Typically, the code is first benchmarked against a configuration for which experimental data is available, agreement is usually found to within a fraction of a percent. Parametric scans are then performed to quantify the behavior of the device across the range of interest. It should be stressed that these devices are usually of sufficient geometrical complexity that analytical solutions are impossible (see Figures 1 and 2, for example). The only alternative to numerical simulation would then be to construct a prototype and perform an experimental study, a process which is usually far too costly and time-consuming.

Fig. 1 (a) ARGUS representation of a portion of a high-power antenna used for ion cyclotron resonance plasma heating in a Tokamak fusion reactor. (b) ARGUS generated contours of the z-component of the electric field at the midplane of the antenna when driven at its fundamental operating frequency.

Fig. 2a ARGUS gridding of a microwave/millimeter wave integrated circuit module housing.

Fig. 2b Electric displacement vectors for the lowest mode of the device.

Accelerator design - The simulation codes traditionally used in the particle accelerator community have either employed simplified physics models (for example, ignoring space charge effects), or else have not been fully three-dimensional. However, modern accelerator designs are indeed three-dimensional, as can be seen from the Lawrence Berkeley Laboratory constant current variable voltage (CCVV) device shown in Figure 3(a). Fully three-dimensional PIC codes are therefore necessary to do a proper analysis.

We are currently using ARGUS to study such devices, with the goal of determining an optimal match between the accelerating system and attainable beam parameters. Preliminary results from such a calculation are shown in Figure 3(b).

Fig. 3 (a) An ARGUS simulation of the LBL CCVV accelerator. The “fingers” are electrostatically charged in a quadrupole configuration so as to provide alternate gradient focusing and defocusing of the beam in the transverse plane as it is accelerated by the voltages on the plates. (b) Comparison between an ARGUS simulation and an envelope calculation for the CCVV accelerator. An x-z projection is on the left, and a y-z projection is on the right. The upper plots show the boundary of the beam as predicted by the envelope code, while the lower plots show the actual beam trajectories as calculated by ARGUS.

V. SUMMARY

Three-dimensional electromagnetic PIC simulation is a mature, well-developed, and cost-effective technique for the analysis of a wide variety of physical devices. As supercomputer-based codes become increasingly better coupled with workstation-based graphical capabilities, these integrated systems have moved to the forefront as everyday design tools. As always, work in the field is ongoing, and we expect to see even more dramatic advances over the next few years.

1 Work supported by the SAIC Independent R&D Program and by the US Government.

2 Also contributing to this work are O. Anderson, S. Brandon, C.F. Chan, C.L. Chang, A. Drobot, K. Ko, M. Kress, A. Mondelli, J. Moura, J. Petillo, and L. Soroka.
APPLICATION OF THE DETERMINISTIC PARTICLE METHOD TO THE WIGNER EQUATION

FRANCIS NIER
CMAP - Ecole Polytechnique
91128 Palaiseau Cedex
FRANCE

Abstract: The Wigner equation that we present here was proposed by physicists as a model for quantum electronic devices in the kinetic regime. The first member of this equation is a drift term while the quantum effects are taken into account in the second member via a Fourier integral operator. We solved this equation by a deterministic particle method for which we proved the convergence.

I - The Wigner equation

The Wigner equation models the motion of electrons in an external electrostatic potential, which we shall decompose in order to describe quantum tunneling effects: the electrons are accelerated by an electric field \( E \) and partially tunnel through a potential barrier given by a real function \( V(x), x \in \mathbb{R}^d \). This equation governs the evolution of a distribution function \( w(x,k,t) \), where \( x \in \mathbb{R}^d, p \in \mathbb{R}^d \) and \( t \in \mathbb{R}^+ \) are respectively the position, the impulsion and the time coordinates:

\[
\begin{align*}
\frac{\partial w}{\partial t} + \frac{p}{m} \frac{\partial w}{\partial x} - qE \frac{\partial w}{\partial p} &= \Theta[V] \cdot w, \\
w(t=0) &= w_0
\end{align*}
\]

The real numbers \( \hbar, m, q \) are respectively the Planck constant, the mass of electron and its charge. The operator \( \Theta[V] \) is a Fourier integral operator defined by

\[
(\Theta[V], w)(x, p, t) = \frac{1}{(2\pi \hbar)^d} \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} \left[ V\left( x + \frac{m}{q} t \right) - V\left( x - \frac{m}{q} t \right) \right] x \wedge w(x', p', t) e^{i(p-p') \cdot \eta} \, dp' \, d\eta
\]

This integral can also be written as a convolution with respect to the impulsion variable:

\[
(\Theta[V], w)(x, p, t) = \int_{\mathbb{R}^d} (q(x, p, p') \wedge w(x', p', t)) \, dp',
\]

with

\[
q(x, p) = \left( \frac{2\pi \hbar}{|p|} \right)^d \Im \left[ e^{2\pi \hbar \frac{x}{|p|^2}} \sqrt{\frac{2\pi}{|p|}} \right]
\]

and

\[
\Theta[V](x) = \frac{1}{(2\pi \hbar)^d} \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} \left[ V\left( x + \frac{m}{q} t \right) - V\left( x - \frac{m}{q} t \right) \right] \, dp' \, d\eta.
\]

A classical semigroup analysis provided in [1] states that the operator \( -\frac{p}{m} \frac{\partial}{\partial x} + qE \frac{\partial}{\partial p} - \Theta[V] \) generates a continuous unitary group in \( L^2(\mathbb{R}^d) \). Thus, if the initial data belongs to \( L^2(\mathbb{R}^d) \), this equation admits a unique solution which satisfies

\[
\|w(t)\|_2 = \|w_0\|_2.
\]

II - The particle method

In order to solve numerically, we used the deterministic particle method introduced by P.A. Raviart and S. Mas-Gallic in [2] - [3]. The exact solution of equation (1.1) is approximated by a linear combination of delta functions:

\[
w(x, v, t) = \sum_i \omega_i w_i(t) \delta(x - x_i(t)) \delta(p - p_i(t))
\]

The motion of the particles in the phase space is given by

\[
\frac{dx_i(t)}{dt} = \frac{p_i(t)}{m}, \quad \frac{dp_i(t)}{dt} = -qE,
\]

while the control volume of the particle \( i, \omega_i \) does not depend on \( t \) and its weight, \( w_i(t) \), is an approximation of \( w(x_i(t), p_i(t), t) \). The weights evolve according to

\[
\frac{dw_i(t)}{dt} = \sum_j \omega_j \xi_j(x_i(t) - x_j(t)) q(x_i(t), p_i(t) - p_j(t)) w_i(t),
\]

that we obtain by first regularizing the integral operator in the \( x \) direction

\[
(\Theta[V], w)(x, p, t) = \int_{\mathbb{R}^d} \xi(x-x') q(x', p-p') \times
\]

and then using a quadrature formula.
The numerical analysis of this method relies on $W^{m,p}$ estimates for the solution of equation (1.1). This estimates can not be obtained by the classical semi-group analysis which yields only (1.6). By using the convolution form of the integral operator, we have proved in [4] the estimate

$$\| w(t) \|_{W^{m,p}} \leq C(t) \| w_1 \|_{W^{m,p}},$$

under the assumption

$$\sup_{|\alpha|+|\beta| \leq m} \left[ \int_{\mathbb{R}^d} |u^\alpha \partial^\beta \psi(u)| \, du \right] < \infty.$$ 

Then the classical arguments of consistence and stability yield the convergence of the method with order $m$.

III - Application to the simulation of a resonant tunneling diode

A model proposed by physicists [5] for the simulation of resonant tunneling diodes relies on the one-dimensional ($d=1$) Wigner equation where the potential $V(x)$ describe a double barrier.

The equation (1.1) has to be solved in a bounded domain $[0,L] \times [p_{\min}, p_{\max}]$ with physical boundary conditions given in [5] at the boundaries $x=0$ and $x=L$, and artificial boundary conditions for $p = p_{\min}$ and $p = p_{\max}$. We chose periodic boundary conditions which are the easiest one to take with the particle method. This model can be improved by coupling with the Poisson equation in order to take into account the electrostatic interaction. Then the electric field depends on the position $E = E(x)$ and is derived from the electrostatic potential. We compute this potential by a Particle In Cell method commonly used for this coupling [6] [7].

The main difficulty of this problem is due to the singularity of the barrier potential of which the Fourier transform does not satisfy the inequality (2.5) even for $m=0$. The particle method was proved to converge in this case but in a very weak sense and without any order. Moreover, the integral (1.3) is an oscillating integral which makes the numerical computations rather heavy. In spite of this, we got results which allowed to justify the simplified models used by physicists.

REFERENCES


Abstract. The aim of the present paper is the presentation of some results about the numerical study of the flow of a polyatomic gas between one evaporating and one totally absorbing plate. Density, velocity and temperature profiles are obtained in the range 0.01≤Kn≤0.1 by solving a kinetic equation by a combination of Monte Carlo and finite difference techniques.

I. INTRODUCTION

The mathematical description of evaporation and condensation phenomena is a classical problem of kinetic theory and a large number of papers has been devoted to the study of its various aspects. Most of the previous investigations were aimed at studying evaporation problems connected to monatomic gases. However, in many situations, which are relevant both for the scientific and practical point of view, one is faced with the evaporation of polyatomic species. The strong evaporation of a polyatomic gas in a half space has been considered by Cerignani [1] who extended Ytrehus's trimodal Ansatz to obtain approximate solutions of a BGK-like kinetic model by a moment method. The aim of the work described in this paper is the modeling of the evaporation of a polyatomic gas by the direct numerical solution of a kinetic equation. The approach has the obvious advantage of requiring no a priori guess of the distribution function, therefore it has a wider range of applicability as compared with the moment method.

The one-dimensional flow resulting from the evaporation/condensation of a polyatomic gas between two parallel plates is considered as model problem.

II. BASIC EQUATIONS

As is well known the mathematical description of the behavior of a polyatomic gas is still an open problem that has been approached in a variety of different ways.

Holway's kinetic equation is

$$\frac{\partial f}{\partial t} + \xi \vec{V} f = \nu_1 f_1 (\xi - f) + \nu_\infty f_\infty$$

(1)

was used as a starting point for this study, since it lends itself to a simple numerical treatment. In Eq. (1) the functions $f_1$ and $f_\infty$ are defined as follows:

$$\nu_1 = \frac{N}{(2\pi RT)^{3/2}} \exp \left[ -\frac{\xi - U}{2RT} \right]$$

(2)

and

$$\nu_\infty = \frac{N}{(2\pi RT)^{3/2}} \exp \left[ -\frac{\xi - U}{2RT} \right] \times$$

$$\left[ \frac{\Gamma(J/2)(kT)^{J/2}}{\Gamma(J/2)(kT)^{J/2}} \right] \exp \left[ -\frac{\xi}{RT} \right]$$

(3)

Eq. (1) approximates the collisional term by the sum of two BGK-like terms. The first of these describes elastic collisions, while the second models anelastic collisions. In Eq. (1) $f(x,t,\xi,c)$ is the distribution function of molecular velocities $\xi$ and internal energy of a gas with $j$ internal (indistinguishable) degrees of freedom. The quantity $n(x,t,c)$ is the number density of molecules, having the internal energy $\xi$, $N(x,t)$ is the total number density, $U(x,t)$ is the mean velocity of the gas, $T(x,t)$ is the translational temperature, $T(x,t)$ is the overall temperature. The collision frequencies have been calculated from the viscosity $\mu(T)$:

$$\nu_v = (1-2\nu_v)\nu_v + 2\nu_v \nu_\infty$$

(4)

being $\nu_v$ the fraction of anelastic collisions. Since the main difficulty lies in the modeling of anelastic collision, it seems reasonable to restrict the use of BGK terms to that process only. A second model was therefore obtained by replacing the first collisional term by the full Boltzmann Equation to describe elastic collisions. The second term was left unchanged.

III. Description of the Numerical Technique

The numerical algorithm is based on a consistent finite difference discretization of Eqs. (1) and (4). The Monte Carlo method is used to evaluate the collision integral at the right hand side of Eq. (4). A generalization of the time-splitting method by Tcheremissine and Aristov was used.

In this work, the region between the plates has been divided into a number of cells of variable size, and
the distribution functions assumed to be constant within each cell. The size of spatial cells was smaller in the regions of stronger gradients. A similar procedure has been adopted to represent the distribution functions in the velocity space. A regular net of nodes is arranged into a finite domain of the velocity space assuming that the distribution function is constant within the cell volumes. The domain has to be chosen in order to contain most of the particles at any stage of the calculations. As far as the discretization of f in the space of internal energy c is concerned, it was convenient to represent the distribution function through a set of values calculated at the nodes of a Gauss integration formula with unit weight function on the interval \([0, E_{max})\). The distribution function was assumed to vanish for \(c > E_{max}\).

IV. RESULTS AND DISCUSSION

The numerical method described above has been used to obtain approximate solutions of Eq. (1) and Eq. (4) for values of the Knudsen number Kn in the range \([0.01, 0.1]\). The Knudsen number considered here is defined as \(\lambda / L\), being \(\lambda\) the mean free path when the gas is in equilibrium with the condensed phase. Only the case of a diatomic molecule (J=2) was considered. The values of the parameter \(z\) were selected in order to have the anelastic collision frequency not too different from the elastic collision frequency. In this way the relaxation phenomena associated with the internal and translational degrees of freedom occur on a scale of comparable magnitude.

The first result of the analysis was that Eq. (1) and Eq. (4) give very close results for the same values of \(Kn\) and \(z\). The largest difference was found in the profiles of the rotational temperature shown in Fig. 1 in the case \(Kn=20, z=0.5\). The difference is due to the fact that the value of the overall collision frequency given by Eq. (5) is slightly higher than the value calculated from viscosity. Therefore, most of the results have been obtained from Holway's model. The effects of the internal degrees of freedom are more evident in the translational temperature profiles (Fig. 2). The density and velocity profiles are very close to those of a monatomic gas. Finally, values of the back scatter fraction are presented in table 1. It was found that, as expected on the ground of the moment method calculations of Ref. 2, a larger fraction is scattered back to the surface in the case of a polyatomic gas.

| Table 1 : Back scatter fraction vs. 1/\(Kn\) |
|----------|------|------|------|------|
| \(z\)    | 1/\(Kn\) | 10   | 20   | 50   | 100  |
| 0        | 0.130  | 0.143 | 0.155 | 0.160 |
| 0.5      | 0.142  | 0.167 | 0.183 | 0.188 |

V. REFERENCES


Abstract: A family of Riemann solvers for the Compressible Euler equations is presented, which preserves physical properties of the flow. They are obtained using a kinetic approach to the Euler Equations.

I. INTRODUCTION

We derive from the kinetic theory of gas, a family of Riemann type solvers for the Compressible Euler Equations that we call Boltzmann solvers. The system we solve is

\[ \begin{align*}
\partial_t \rho + \nabla \cdot (\rho u) &= 0, \\
\partial_t \rho u + \nabla \cdot (\rho u u) + \partial_y p &= 0,
\end{align*} \]

where \( \rho, u, p \) are the density, velocity, and pressure, respectively. The classical choice of \( \chi(w) \) is the Maxwellian \( \chi(w) = \exp(-w^2/22) / \sqrt{2\pi} \) (see Deshpande for instance). In order to preserve (i)-(iii) we propose to use other choices of \( \chi \), which also give better numerical results.

II. BOLTZMANN SCHEMES.

We present another method, which allows to possibly take into account the corners effect (dependency of \( F_{i+1/2,j} \) on \( U_{i+1,j+1} \) for instance). This is achieved using a kinetic approximation to Euler Equations

\[ \begin{align*}
\partial_t f + v \cdot \nabla f &= 0 \quad \text{on } (n\Delta t, (n+1)\Delta t) \\
\{ (n\Delta t, z, v) &\rightarrow \rho^n f^n \chi((v-u^n)/\sqrt{\rho^n}) \\
\text{where } (1, w_1^2, w_2^2) \chi(w) dw &= (1, 1, 1) \text{ and } 0 \leq \chi(w) \leq \chi(-w). 
\end{align*} \]

Indeed polynomial choices of \( \chi \) lead to integrate polynomials in order to get \( F_{i+1/2,j} \). The problem we address is to find finite volume approximations of the above system under the form

\[ 
U_{ij}^{n+1} - U_{ij}^n + \sigma_x (F_{i+1/2,j} - F_{i-1/2,j}) + \sigma_y (G_{i+1/2,j} - G_{i-1/2,j}) = 0,
\]

where \( \sigma_x = \Delta t / \Delta x, \sigma_y = \Delta t / \Delta y \), and \( F_{i+1/2,j} \) for example is an appropriate approximation of

\[ \begin{align*}
\rho_{ij}^{n+1} u_{ij}^{n+1} + \rho_{ij}^{n+1} p_{ij}^{n+1} u_{ij}^{n+1} (x_{i+1/2}, y_j) \text{ such that, under a CFL condition,} \\
(i) \rho_{ij}^{n+1} \geq 0, \quad T_{ij}^{n+1} \geq 0, \\
(ii) \text{the entropy inequality holds} \\
(iii) \text{the maximum principle on the entropy holds}
\end{align*} \]

\[ \min(S_{ij}, \min_{k \in \{x_{i+1/2}, y_j\}} S_{mk}) \leq \max(S_{ij}, \min_{k \in \{x_{i+1/2}, y_j\}} S_{mk}) \]

where \( S \) is an entropy of Euler Equations for instance

\[ S = \rho (T-1)/T. \]

These properties can be realized using an exact solver (Godunov) i.e. by computing the exact value \( U_{i+1,j} \) at the mesh interface.

III. NUMERICAL RESULTS.

The following figure presents, for a 2D stationary reflection problem, the improvement obtained using the exact solver, compared to the simplified solver described above.
A MACSYMA PROGRAM FOR THE HIROTA METHOD

WILLY HEREMAN AND WUNING-ZHUANG
Department of Mathematical and Computer Sciences
Colorado School of Mines
Golden, CO 80401, USA

Abstract - Hirota's method for finding soliton solutions of nonlinear evolution and wave equations is briefly discussed and illustrated. A MACSYMA program that automatically carries out the lengthy algebraic computations is included.

I. INTRODUCTION

Hirota's direct method [1,2] allows to construct exact soliton solutions of nonlinear evolution and wave equations. The lengthy but straightforward calculations inherent to this method can easily be performed with any symbolic manipulation program. In this paper we present a sample program in MACSYMA to illustrate the symbolic calculation of one, two and three soliton solutions of well-known nonlinear PDEs such as the Korteweg-de Vries, the Boussinesq, the Kadomtsev-Petviashvili, the Sawada-Kotera and the shallow water wave equations [1-5].

II. THE HIROTA-METHOD

Hirota's method requires:
(i) a clever change of dependent variable,
(ii) the introduction of a novel differential operator,
(iii) a perturbation expansion to solve the resulting bilinear equation.
Details about the method can be found in almost any book on soliton theory [1-4], here we merely outline the procedure.

Our leading example is the Korteweg-de Vries equation [1-4],

\[ u_t + 6uu_x + u_{xx} = 0 \]  \hspace{1cm} (1)

Substitution of

\[ u(x,t) = 2^{\theta/2} \ln f(x,t) \]  \hspace{1cm} (2)

into (1) and one integration with respect to \( x \) yields,

\[ f_{tt} - 6f_t + f f_{xx} - 4f_x f_x + 3f_x^2 = 0 \]  \hspace{1cm} (3)

This quadratic equation in \( f \) can then be written in bilinear form,

\[ B(f,1) \frac{D^2}{Dx^2} (f+f) (f-f) = 0 \]  \hspace{1cm} (4)

where the new operator is given by

\[ D^2 = \frac{D}{Dx} \frac{D}{Dt} \left( \frac{D}{Dx} - \frac{D}{Dt} \right) \]  \hspace{1cm} (5)

Introducing a book keeping parameter \( c \), we look for a solution

\[ f = 1 + \sum_{n=1}^{\infty} c^n f_n \]  \hspace{1cm} (6)

Substituting (6) into (4) and equating to zero the powers of \( c \), yields

\[ O(c^0) : \ B(1-1) = 0 \]  \hspace{1cm} (7)
\[ O(c^1) : \ B(1+f_1+f_1) = 0 \]  \hspace{1cm} (8)
\[ O(c^2) : \ B(1+f_2+f_2+f_2) = 0 \]  \hspace{1cm} (9)
\[ O(c^3) : \ B(1+f_3+f_3+f_3+f_3) = 0 \]  \hspace{1cm} (10)
\[ O(c^4) : \ B(1+f_4+f_4+f_4+f_4+f_4) = 0 \]  \hspace{1cm} (11)
\[ O(c^n) : \ B(\sum_{j=0}^{n} f_j f_{n-j}) = 0 \]  \hspace{1cm} (12)

This scheme is general whatever the explicit expression of the bilinear operator \( B \) is. For the KdV equation \( B \) is given in (4).

If the original PDE admits a \( N \)-soliton solution then (6) will truncate at level \( n = N \) provided \( f_1 \) is the sum of precisely \( N \) simple exponential terms. For simplicity, consider the case of a three soliton solution (\( N = 3 \)). Then,

\[ f_1 = \sum_{i=1}^{3} \exp(\theta_i) = \sum_{i=1}^{3} \exp \left( \frac{k_i x - \omega_i t + \delta_i} {t} \right) \]  \hspace{1cm} (13)

where \( k_i, \omega_i \), and \( \delta_i \) are constants. Of course, (7) is trivially satisfied, whereas (8) determines the dispersion law,

\[ \omega_i = k_i^2 \]  \hspace{1cm} (14)

The terms generated by \( B(f_1, f_1) \) in (9) justify the choice

\[ f_2 = a_{12} \exp(\theta_1 + \theta_2) + a_{13} \exp(\theta_1 + \theta_3) + a_{23} \exp(\theta_2 + \theta_3) \]  \hspace{1cm} (15)

\[ = a_{12} \exp \left( (k_1 + k_2) x - (\omega_1 + \omega_2) t + \delta_1 + \delta_2 \right) \]
\[ + a_{13} \exp \left( (k_1 + k_3) x - (\omega_1 + \omega_3) t + \delta_1 + \delta_3 \right) \]
\[ + a_{23} \exp \left( (k_2 + k_3) x - (\omega_2 + \omega_3) t + \delta_2 + \delta_3 \right) \]

and (9) allows to calculate the constants \( a_{12}, a_{13} \) and \( a_{23} \). With (14) one obtains

\[ a_{ij} = \frac{(k_i - k_j)^2}{(k_i + k_j)^2} \]  \hspace{1cm} (16)

Then, \( B(f_1, f_2, f_2) \) in (10) motivates the particular solution

\[ f_3 = b_{123} \exp(\theta_1 + \theta_2 + \theta_3) \]
\[ = b_{123} \exp \left( (k_1 + k_2 + k_3) x - (\omega_1 + \omega_2 + \omega_3) t + \delta_1 + \delta_2 + \delta_3 \right) \]  \hspace{1cm} (17)

and one calculates

\[ b_{123} = a_{12} a_{13} a_{23} = \frac{(k_1 - k_2)^2 (k_2 - k_3)^2 (k_3 - k_1)^2}{(k_1 + k_2)^2 (k_1 + k_3)^2 (k_2 + k_3)^2} \]  \hspace{1cm} (18)

Subsequently, (11) allows to verify that indeed \( f_3 = 0 \). In the sixth equation of the scheme \( B(f_2, f_2, f_2) \) should equal zero in order to assure that \( f_4 = 0 \). If so, it will be possible to take \( f_1 = 0 \) for \( i > 0 \). Finally setting \( c = 1 \) in (6), we obtain

\[ f = 1 + \exp(\theta_1 + \exp(\theta_2 + \exp(\theta_3 + \exp(\theta_1 + \theta_2) + \exp(\theta_2 + \theta_3) + \exp(\theta_1 + \theta_3) \]
\[ + \exp(\theta_1 + \theta_2 + \theta_3) \]  \hspace{1cm} (19)

which upon substitution in (2) generates the well-known three soliton solution of (1).

The construction of \( N \)-soliton solutions [1-6] with \( N \geq 3 \) is tedious and the necessary algebraic simplifications and factorizations are bound to fail if carried out by hand. Hence the need for a symbolic program that relieves us of the elaborate calculations.

III. MACSYMA PROGRAM FOR THE HIROTA METHOD

This preliminary program calculates the one, two and three soliton solutions of a fairly simple completely integrable PDE, provided it can be transformed into a single bilinear equation for the new variable \( f \). The program is written in such a way that the extension for the \( N \)-soliton is straightforward. The structure should also allow to 'translate' it into the language of e.g. MAPLE, MATHEMATICA, or REDUCE. The user must select the value of \( N \) and also provide the bilinear operator \( B \) for the PDE.
For the Boussinesq equation [1-6], $u_{21} - u_{22} - 3(u^2)_{22} - u_{23} = 0$, one has again (2) and:

$$B(f,g) := D_x(0,2)[f,g] - D_x(2)[f,g] - \frac{D_y[4]}{4}[f,g].$$

The results then are $\omega_i = -k_i \sqrt{1 + k_i^2}$, $i = 1, 2, 3$, and

$$a_{ij} = \frac{1}{\sqrt{1 + k_i^2}} \left[ 1 + k_i^2 - 2k_i^2 + 3k_i k_j - 2k_i - 1 \right], \quad i, j = 1, 2, 3.$$  \eqref{21}

The program did also determine the explicit form of $b_{122} = a_{12} a_{13} a_{23}$.  

For the Sawada-Kotera equation [2,5,6],

$$u_{11} + 45u_{22} + 15u_{22} + 15u_{32} + u_{x3} = 0,$$

one has (2) and $B(f,g) := D_x[1,1][f,g] + D_x[0][f,g]$. Furthermore, $\omega_i = k_i^2$, $i = 1, 2, 3$. The coefficients are given by

$$a_{ij} = \frac{(k_i - k_j)^2 (k_i^2 - k_j k_i + k_j^2)}{(k_i^2 - k_j^2 k_i - 2k_j + k_i)^2}, \quad i, j = 1, 2, 3, \quad \eqref{22}$$

$$b_{122} = a_{12} a_{13} a_{23}. \quad \eqref{23}$$

For the shallow water wave equation [5],

$$u_{22} + 3u_{44} - u_x u_{44} - u_{22} - u_2 = 0$$

and $B(f,g) := D_x[1][f,g] + D_x[0][f,g]$.

The program calculated that

$$\omega_i = \frac{k_i}{(1 + k_i)(1 - k_i)}, \quad i = 1, 2, 3, \quad \eqref{24}$$

$$a_{ij} = \frac{(k_i - k_j)^2 (k_i^2 - k_j k_i + k_j^2 - 3)}{(k_i^2 - k_j^2 k_i - 2k_j + k_i)^2}, \quad i, j = 1, 2, 3. \quad \eqref{25}$$

The program could not determine $b_{122} = a_{12} a_{13} a_{23}$ due to the large number of terms in $f_2$.

Hirota's bilinear operators used in these examples, $D_x[m,n][f,g]$, $D_x[n][f,g]$, and $D_y[n][f,g]$ are defined in the program itself.

REFERENCES


A DIFFERENTIAL-DIFFERENCE EQUATION FOR
HIGHER NONLINEAR SCHRODINGER EQUATION

Thal R. Tuha
Computer Science Department
University of Georgia
415 GSRC
Athens, GA 30602 U.S.A.

Abstract - A new differential difference equation is obtained which has as its limiting form a higher nonlinear Schrödinger (HNLS) equation. This new equation is constructed by methods related to the inverse scattering transform (IST) and can be used as a numerical scheme for the HNLS equation.

1. INTRODUCTION

In 1975 Ablowitz and Ladik proposed a new discrete eigenvalue problem, an appropriate generalization of a discretized version of the eigenvalue problem of Zakharov and Shabat, as a basis for generating solvable discrete equations [1]. They derived discrete (differential-difference) versions of the nonlinear Schrödinger (NLS), Korteweg-de Vries (KdV), modified KdV, and "sine-Gordon" equations.

In 1984 Tuha and Ablowitz derived differential-difference and partial difference equations for KdV, and MKdV equations [2]. In this paper a differential-difference equation is obtained which has as its limiting form the HNLS equation

\[ \dot{q}_i = q_{xxx} + 8q_1q_i^2 + 4q_iq_{i+1}^2 + 6q_i^3 (q_2)^2 + 2q^2 q_{i+1}^3 + 6q_i q_{i+1} \]

This differential-difference equation can be used as a numerical scheme for the HNLS equation (1).

II. DERIVING NONLINEAR DIFFERENTIAL-DIFFERENCE EQUATION

The key step in obtaining differential-difference equations which can be solved by the inverse scattering transform is to make an association between the nonlinear evolution equation and a linear eigenvalue (scattering) problem. To find a nonlinear differential-difference equation associated with the HNLS equation, it is essential to use (a) a suitable eigenvalue problem e.g.,

\[ V_{1n+1} = z V_{1n} + Q_n(t) V_{2n} \]
\[ V_{2n+1} = \frac{1}{z} V_{2n} + R_n(t) V_{1n} \]

where \( z \) is the eigenvalue and the potentials \( R_n, Q_n \) are defined on the spacelike interval \( 0 < t < \infty \) and the time \( t > 0 \), and (b) the associated time dependence,

\[ V_{11} = A_n V_{1n} + B_n V_{2n} \]
\[ V_{22} = C_n V_{1n} + D_n V_{2n} \]

where the functions \( A_n, B_n, C_n, D_n \) depending in general on the potentials. The equations for determining the sets \( A_n, B_n, C_n, D_n \) and hence the evolution equation are obtained by requiring the eigenvalue \( z \) to be time invariant \( \frac{\partial z}{\partial t} = 0 \) and by forcing the consistency

\[ \frac{\partial}{\partial t} (E V_{1i}) = E (V_{1i}), \quad i = 1, 2 \]

where \( E \) is the shift operator defined by \( E V_{1i} = V_{1i+1} \), \( i = 1, 2 \). Performing the operations indicated in (4) results in four equations which are given by

\[ z (A_n A_n) - C_n Q_n + R_n B_{n+1} = 0 \]
\[ \frac{1}{z} B_{n+1} = z B_n + Q_n (A_{n+1} - D_n) = Q_{n-1} \]
\[ \frac{1}{z} C_{n+1} - R_n A_n + R_n D_n - \frac{1}{z} C_n = R_{n-1} \]

(5)

\[ \frac{1}{z} (A_n D_n) + C_n Q_n - R_n B_n = 0 \]

where \( \Delta_n A_n = A_{n+1} - A_n, \) etc.

Using the ideas in [1,2,3], the coefficients for the time dependence of the eigenfunctions are expanded as follows:

\[ A_n = \sum_{k=1}^{2} z^{2k} A_n^{(2k)}, B_n = \sum_{k=1}^{2} z^{2k-1} B_n^{(2k)}, \]
\[ C_n = \sum_{k=1}^{2} z^{2k-1} C_n^{(2k)}, D_n = \sum_{k=1}^{2} z^{2k} D_n^{(2k)} \]

(6)

With the expanded form of \( A_n, B_n, C_n, D_n, \) (5) yields a system of coupled equations in eighteen unknowns corresponding to equating powers of \( z, z^2, z^3, z^4, \ldots, z, z^{-1} \), all of which must be independently satisfied. Carrying out the algebra we find the values of \( A_n^{(0)}, B_n^{(0)}, C_n^{(0)}, D_n^{(0)} \) in terms of the potentials [2]. The remaining two equations are the evolution equations. For the special case associated with the NLS and HNLS equation we let \( R_n = -Q_n \) (where \( Q_n^* \) is the complex conjugate of \( Q_n \)), then the remaining two coupled equations are consistent under the conditions:

\[ A_n^{(0)} - D_n^{(0)} = D_n^{(2)} + A_n^{(2)} = \alpha \]
\[ A_n^{(0)} - D_n^{(0)} = D_n^{(2)} + A_n^{(2)} = \beta \]

(7)

\[ A_n^{(0)} - D_n^{(0)} = D_n^{(2)} + A_n^{(2)} = \gamma \]

with \( \alpha = -\alpha^*, \beta = -\beta^*, \) and \( \gamma = -\gamma^* \) and the nonlinear differential-difference equation is

\[ Q_{n+1} = (1 + |Q_n|^2) (Q_{n+1} + Q_{n+1}^*) + \alpha (Q_{n+2} + Q_{n-2}) + Q_{n+2} + Q_{n+2} \]
\[ + Q_{n-2} + Q_{n-2}^* + Q_{n-2}^* + Q_n^2 + Q_n^2 + Q_n^2 + Q_n^2 + \]
\[ + Q_n (Q_{n+1}^* + Q_{n+1}^*) \]

(8)
Let \( Q_n = \Delta x q_n \), taking the limit as \( \Delta x \to 0 \) in (8) and by a proper choice of the constants

(a) \( \alpha = \frac{-i}{12(\Delta x)^2}, \ \beta = \frac{-16i}{12(\Delta x)^2}, \) and \( \gamma = \frac{30i}{12(\Delta x)^2} \) yields the NLS equation

\[
\partial_t q = q_{xx} + 2|q|^2q; \quad (9)
\]

(b) \( \alpha = \frac{-i}{(\Delta x)^2}, \ \beta = \frac{4i}{(\Delta x)^2}, \) and \( \gamma = \frac{-6i}{(\Delta x)^4} \) yields the HNLS equation (1).

### III. CONCLUSION

The differential-difference equations derived in this paper have as limiting forms the NLS and HNLS equations. These equations can be used as numerical schemes for the associated nonlinear evolution equations. These IST schemes have a number of special properties [2].

### IV. ACKNOWLEDGEMENTS

This research has been supported in part by the Department of Energy Grant No. DE-FG09-90ER25088.

### REFERENCES


Nonlinear Evolution Equations and Painlevé Test

W. H. Steeb and N. Euler

Department of Applied Mathematics and Nonlinear Studies
Rand Afrikaans University, Johannesburg 2000, South Africa

Abstract: The Painlevé test is a powerful tool [1-6] in the study of nonlinear evolution equations. We can study the integrability of ordinary and partial differential equations. Exact solutions can be constructed. Bäcklund transformations ... can be derived within this approach. By making use of the Painlevé test the construction of Lie Bäcklund symmetries is straightforward. It also plays a fundamental role in the investigation of the chaotic behaviour for ordinary differential equations. We apply the Painlevé test to the anharmonic oscillator, the semiclassical Jaynes-Cummings model, the energy eigenvalue level motion equation, the Kostov-Petviashvili equation, the nonlinear Klein Gordon equation and the self-dual Yang-Mills equation and its connection with the Yang-Mills equation.

First we discuss ordinary differential equations. The ordinary differential equations are extended into the complex domain.

Definition: An ordinary differential equation is said to have the Painlevé property when every solution is single valued, except at the fixed singularities of the coefficients. That is, the Painlevé property requires that the movable singularities are no worse than poles.

Theorem: A necessary condition that an n-th order ordinary differential equation of the form \(dw/dz = g(w)\) where \(g\) is rational in \(w\) has the Painlevé property is that there is a Laurent expansion

\[w_k(z) = (z - z_1)^{m_k} \sum_{j=0}^{\infty} a_{kj}(z - z_1)^j\]

with \(n - 1\) arbitrary expansion coefficients (besides the pole position which is arbitrary).

In the following we give two examples where we apply the Painlevé test to find solutions.

Example 1. The semiclassical Jaynes-Cummings model is given by

\[
\begin{align*}
\frac{dS_1}{dt} &= -S_2, & \frac{dS_2}{dt} &= S_1 + S_3 E, & \frac{dS_3}{dt} &= -S_2 E, & \frac{dE}{dt} &= \alpha S_1 \end{align*}
\]

and \(m_1 = -3, m_2 = -4, m_3 = -4\) and \(m_1 = -1\). For the expansion coefficients we obtain \(S_{1,0} = \frac{1}{2}, S_{2,0} = \frac{1}{2}, S_{3,0} = -\frac{1}{2}, E_0 = i\). From the dominant behaviour we conclude that the system with the dominant terms is scale invariant under \(t \mapsto e^{-i \tau t}, S_1 \mapsto e^3 S_1, S_2 \mapsto e^3 S_2, S_3 \mapsto e^3 S_3, E \mapsto e^3 E\). Next we determine the resonances and the Kowalevski exponents. Inserting the ansatz \((j = 1, 2, 3)\)

\[
\begin{align*}
S_j(t) &= S_j(0)(t - t_0)^{m_j} + A(t - t_0)^{m_j+1} + B(t - t_0)^{m_j+2} + \cdots, \\
E(t) &= E_0(t - t_0)^{-1} + D(t - t_0)^{-1+1} + \cdots
\end{align*}
\]

into the system with the dominant terms we find the resonances

\[-1, 4, 8, 3/2 \pm i \sqrt{3}/2\]

The Kowalevski exponents can be found from the variational equation. We find that the resonances and the Kowalevski exponents coincide. The two Kowalevski exponents 1 and 8 can be related to first integrals. We obtain \(I_1 = S_1^2 + S_3^2\) and \(I_2 = \alpha S_3 - \alpha S_2 E + (dE/dt)/2\) since \(I_1(e^{\tau} S_1, e^{\tau} S_2, e^{\tau} S_3, e^{\tau} E) = e^\tau (S_1^2 + S_3^2), I_2(e^{\tau} S_1, e^{\tau} S_2, e^{\tau} S_3, e^{\tau} E) = e^\tau (\alpha S_3 - \alpha S_2 E + (dE/dt)/2)\). Using the first integrals for the system with the dominant terms we find on inspection that the first integrals for the Jaynes-Cummings model are given by \(I_1 = S_1^2 + S_3^2 + S_2^2 + 4(1/2 + 1/2 \cdots) + (dE/dt)/2\). From the Painlevé test we find that the Jaynes-Cummings model admits a Laurent expansion of the form \((j = 1, 2, 3)\)

\[
S_j(t) = \sum_{i=0}^{\infty} S_{ji}(t - t_0)^{m_{ji}}, \quad E(t) = \sum_{i=0}^{\infty} E_{ji}(t - t_0)^{-1}
\]

with three arbitrary constants (including \(t_0\)). The expansion coefficients are determined by a recursion relation. In particular we find \(S_{1,1} = S_{2,1} = S_{3,1} = E_1 = 0\). This local expansion is not the general solution (owing to the complex resonances) which requires five arbitrary constants. We now construct an exact solution from the Laurent expansions (9) through (12). Let \(k\) be the modulus of the elliptic functions \(sn(z, k), cn(z, k)\) and \(dn(z, k)\). We define \(k^2 = f_4(1 - k^2)^{-1/2}(1 - k^2)^{1/2} dt\) where \(k^2 = 1 - k^2\). By the addition-theorem of the Jacobian elliptic functions, we have \(sn(z + iK', k) = 1/ksn(z, k)\). Similarly \(cn(z + iK', k) = -i/kdn(z, k)/sn(z, k), dn(z + iK', k) = -i/cn(z, k)/sn(z, k)\). For points in the neighbourhood of the point \(z = 0\), the function \(sn(z, k)\) can be expanded by Taylor's theorem in the form \(sn(z, k) = sn(0, k) + zsn'(0, k) + 1/2z^2sn''(0, k) + \cdots \) where accents denote derivatives. ce \(sn(0, k) = 0, sn'(0, k) = 1, sn''(0, k) = 0, sn'''(0, k) = -(1 + k^2)\) etc. the expansion becomes

\[sn(z + iK', k) = \frac{1}{k} \frac{1}{\sin(z, k)} = \frac{1}{k^2} \frac{1}{6k} z + \frac{1}{6k^2} (1 + k^2)^2 z^2 + \cdots\]

Therefore \(cn(z, k) = (1 - sn^2 z)^{1/2} = 1 - 1/2z^2 + \cdots \) and \(dn(z, k) = (1 - k^2sn^2 z)^{1/2} = 1 - 1/2k^2z^2 + \cdots \). Consequently

\[sn(z + ik', k) = \frac{1}{k} \frac{1}{\sin(z, k)} = \frac{1}{k^2} \frac{1}{6k} z + \frac{1}{6k^2} (1 + k^2)^2 z^2 + \cdots\]

Similarly, we find that \(cn(z + ik', k) = \frac{1}{k} \frac{1}{\sin(z, k)} + \frac{2k^2}{6k} z^2 + \cdots \) and

\[dn(z + ik', k) = -\frac{1}{k} + \frac{2k^2}{6k^2} z^2 + \cdots \]

It follows that at the point \(z = iK'\) the functions \(sn(z, k), cn(z, k)\) and \(dn(z, k)\) have simple poles, with the residues \(1/k, -i/k, -i, \text{ respectively}\). We can focus our attention to the quantity \(E\) since \(S_1\) can be derived
The general solution contains three free parameters $I_1$, $I_2$, and $\lambda_0$ which are determined from the initial values $E(\lambda = 0)$, $p(\lambda = 0)$, and $V(\lambda = 0)$.

Next we consider partial differential equations. Suppose that there are $n$ independent variables, and that the system of partial differential equations has coefficients that are holomorphic on $C^n$. We cannot simply require that all the solutions of this system be meromorphic on $C^n$, since arbitrarily nasty singularities can occur along characteristic hypersurfaces. The following definition [2] of the Painlevé property avoids this problem. If $S$ is a holomorphic non-characteristic hypersurface in $C^n$, then every solution that is holomorphic on $C^n \backslash S$ extends to a meromorphic solution on $C^n$.

In other words, if a solution has a singularity on a non-characteristic hypersurface, then that singularity is a pole and nothing worse. A slightly weaker form of the Painlevé property was formulated by Weiss et al [1]. It involves looking for solutions $\phi$ of the system of partial differential equations in the form

$$u = \phi^{-\alpha} \sum_{n=0}^{\infty} u_n \phi^n$$

where $\phi$ is a holomorphic function whose vanishing defines a non-characteristic hypersurface. Substituting this series into the partial differential equations yields conditions on the number $\alpha$ and recursion relations for the functions $u_n$. The requirement is that $\alpha$ should turn out to be a non-negative integer, and the recursion relations should be consistent, and that the series expansion should contain the correct number of arbitrary functions (counting $\phi$ as one of them).

It has been found that integrable equations satisfy this weaker form (perhaps after a change of variables), whereas non-integrable equations fail it. To establish Painlevé property is more difficult (the Painlevé property implies the weaker form, but the reverse implication need not hold in general). However, it seems that in practice, the weaker form is sufficient to ensure integrability.

The Painlevé property seems to be a useful indicator of integrability or solvability, or both. Integrable here means there exists a nontrivial Lax Pair for the system.

**Example 1:** The Kadomtsev-Petviashvili equation is given by

$${\partial^2 u \over \partial t \partial x} + {\partial^2 u \over \partial z^2} + {\partial^4 u \over \partial x^2 \partial y^2} + 3\sigma^2 {\partial^2 u \over \partial y^2} = 0$$

where $\sigma^2 = \pm 1$. The Kadomtsev Petviashvili equation is a completely integrable soliton equation. The Kadtovtsev Petviashvili equation has the Painlevé property. We consider the generalized variable-coefficient Kadomtsev Petviashvili equation

$${\partial^2 u \over \partial t \partial x} + {\partial^2 u \over \partial z^2} + {\partial^4 u \over \partial x^2 \partial y^2} + a(y,t) {\partial u \over \partial y} + b(y,t) {\partial^2 u \over \partial y^2} = 0$$

where

$$a(y,t) = \alpha(y,t); b(y,t) = \beta(y,t).$$

In this case, we should require that $\alpha(y,t)$ and $\beta(y,t)$ be such that the Painlevé property is satisfied.

**Example 2:** In the study of energy level motion of the Hamiltonian system we arrive for a two-level system at the following system:

$$\begin{align*}
E &= \epsilon_1 \xi_1 \xi_2 \phi_1 \phi_2, \\
\xi_1 &= \epsilon_1 \xi_1 \phi_1 \phi_2, \\
\xi_2 &= \epsilon_2 \xi_2 \phi_1 \phi_2.
\end{align*}$$

The quantities $\xi_1, \xi_2$ and $\phi$ can now easily be found from the radial equations.

**Example 3:** The general solution contains three free parameters $I_1$, $I_2$, and $\lambda_0$ which are determined from the initial values $E(\lambda = 0)$, $p(\lambda = 0)$, and $V(\lambda = 0)$.

Next we consider partial differential equations. Suppose that there are $n$ independent variables, and that the system of partial differential equations has coefficients that are holomorphic on $C^n$. We cannot simply require that all the solutions of this system be meromorphic on $C^n$, since arbitrarily nasty singularities can occur along characteristic hypersurfaces. The following definition [2] of the Painlevé property avoids this problem. If $S$ is a holomorphic non-characteristic hypersurface in $C^n$, then every solution that is holomorphic on $C^n \backslash S$ extends to a meromorphic solution on $C^n$.

In other words, if a solution has a singularity on a non-characteristic hypersurface, then that singularity is a pole and nothing worse. A slightly weaker form of the Painlevé property was formulated by Weiss et al [1]. It involves looking for solutions $\phi$ of the system of partial differential equations in the form

$$u = \phi^{-\alpha} \sum_{n=0}^{\infty} u_n \phi^n$$

where $\phi$ is a holomorphic function whose vanishing defines a non-characteristic hypersurface. Substituting this series into the partial differential equations yields conditions on the number $\alpha$ and recursion relations for the functions $u_n$. The requirement is that $\alpha$ should turn out to be a non-negative integer, and the recursion relations should be consistent, and that the series expansion should contain the correct number of arbitrary functions (counting $\phi$ as one of them).

It has been found that integrable equations satisfy this weaker form (perhaps after a change of variables), whereas non-integrable equations fail it. To establish Painlevé property is more difficult (the Painlevé property implies the weaker form, but the reverse implication need not hold in general). However, it seems that in practice, the weaker form is sufficient to ensure integrability.

The Painlevé property seems to be a useful indicator of integrability or solvability, or both. Integrable here means there exists a nontrivial Lax Pair for the system.

**Example 1:** The Kadomtsev-Petviashvili equation is given by

$${\partial^2 u \over \partial t \partial x} + {\partial^2 u \over \partial z^2} + {\partial^4 u \over \partial x^2 \partial y^2} + 3\sigma^2 {\partial^2 u \over \partial y^2} = 0$$

where $\sigma^2 = \pm 1$. The Kadomtsev Petviashvili equation is a completely integrable soliton equation. The Kadtovtsev Petviashvili equation has the Painlevé property. We consider the generalized variable-coefficient Kadomtsev Petviashvili equation

$${\partial^2 u \over \partial t \partial x} + {\partial^2 u \over \partial z^2} + {\partial^4 u \over \partial x^2 \partial y^2} + a(y,t) {\partial u \over \partial y} + b(y,t) {\partial^2 u \over \partial y^2} = 0$$

where

$$a(y,t) = \alpha(y,t); b(y,t) = \beta(y,t).$$

In this case, we should require that $\alpha(y,t)$ and $\beta(y,t)$ be such that the Painlevé property is satisfied.

**Example 2:** In the study of energy level motion of the Hamiltonian system we arrive for a two-level system at the following system:

$$\begin{align*}
E &= \epsilon_1 \xi_1 \xi_2 \phi_1 \phi_2, \\
\xi_1 &= \epsilon_1 \xi_1 \phi_1 \phi_2, \\
\xi_2 &= \epsilon_2 \xi_2 \phi_1 \phi_2.
\end{align*}$$

The quantities $\xi_1, \xi_2$ and $\phi$ can now easily be found from the radial equations.
where \(a(y, t), b(y, t), c(y, t), d(y, t), e(y, t)\) are analytic functions. In order that the generalized Kadomtsev Petviashvili equation satisfy the Painlevé property the function \(a, b, c, d, e\) have to satisfy the system of equations
\[
\begin{align*}
\frac{\partial a}{\partial t} + 2a^2 + \frac{\partial a}{\partial y} + \frac{1}{2} \frac{\partial c}{\partial x} \cdot \frac{\partial c}{\partial y} - \frac{\partial^2 c}{\partial y^2} &= 0 \\
\frac{\partial c}{\partial t} + 4ac - 2\frac{\partial a}{\partial t} + d \frac{\partial c}{\partial y} &= 0 \\
(2b - \frac{\partial c}{\partial y})c &= 0, \\
(2b - \frac{\partial c}{\partial y}) &= 0.
\end{align*}
\]

These equations are necessary and sufficient conditions for the generalized Kadomtsev Petviashvili equations to be transformable into the Kadomtsev Petviashvili equation, provided that \(c \neq 0\).

If \(c = 0\), the equation may be transformed into the Korteweg de Vries equation provided that \(b = 0\) and \(\partial a/\partial t + \partial a/\partial y + 2a^2 = 0\).

**Example 1.** Consider the nonlinear Klein Gordon equation
\[
\frac{\partial^2 u}{\partial t^2} = V_0 \nabla^2 u + \phi(u, \nabla u).
\]
Inserting the expansion
\[
u = e^{m \sum_{j=0}^{\infty} v_j \psi^j}
\]
we find \(m = -1, v_0 = 2\phi \psi \phi_0\) and
\[
- \phi_0 \psi v_0 - \phi_0 \psi \phi_0 = 3 \psi \psi_1 v_1
\]
\[
\psi v_0 = 3 \psi v_1 + 3 \nu \nu_1
\]
\[
2 \phi_0 \psi \psi_2 + \phi_0 \psi \phi_2 + \phi_0 \psi \phi_2 = 2 \psi \psi_2 + 6 \nu \nu_1 v_1.
\]

At the resonance \(r = 4\) we obtain
\[
2 \phi_0 \psi_2 + 2 \phi_0 \psi_2 + 2 \phi_0 \nu_2 + \nu_2 = 6 \nu v_2 + 3 \nu \nu_2 + 3 \nu \nu_2.
\]

Inserting the equations for the coefficients into the above equation for the resonance we obtain a "huge" partial differential equation for \(f(\psi)\). If \(f\) satisfies this condition, then the expansion coefficient \(v_0(\xi, \eta)\) is arbitrary. The nonlinear Klein Gordon equation admits the symmetry vector fields \(\partial/\partial \xi, \partial/\partial \eta, \partial/\partial \xi - \partial/\partial \eta, \partial/\partial \xi + \partial/\partial \eta - \partial/\partial \eta\).

Next we construct similarity ansätze:

(1) From \(\partial/\partial \eta\) and \(\partial/\partial \xi\) we find \(s = c_1 \xi + c_2 \eta\) and \(v(\eta, \xi) = f(s)\). We obtain \(d^2 f/ds^2 = f'(c_1 \xi)\). This equation passes the Painlevé test. Therefore \(f(\eta, \xi) = c_1 \xi + c_2 \xi\) satisfy the condition on \(f\).

(2) The symmetry generator \(\xi \partial/\partial \xi - \eta \partial/\partial \eta\) leads to the similarity ansatz \(v(\eta, \xi) = f(s)\) with \(s = \eta \xi\). It follows that \(d^2 f/ds^2 = f'(c_1 \xi)\). This equation does not pass the Painlevé test. This is in agreement that \(f(\psi(\xi, \eta)) = \xi \psi \xi \psi \xi\) does not satisfy the conditional equation on \(f\).

(3) The symmetry generator \(\xi \partial/\partial \xi + \eta \partial/\partial \eta - \partial/\partial \psi\) leads to the similarity ansatz \(v(\eta, \xi) = f(s)\) with \(s = \eta \xi\). It follows that \(d^2 f/ds^2 + 2 d\psi/ds + f'/s = 0\). This equation passes the Painlevé test. This is in agreement that \(f(\eta, \xi) = \eta \xi\psi \psi \psi \psi \psi\psi\) satisfy the conditional equation on \(f\).

**Example 3.** We now consider the self-duality Yang Mills equations. Let \(G\) be a Lie group (the 'gauge group') and \(g\) it Lie algebra. A gauge potential (connection) \(A\) is a \(g\)-valued 1-form on \(\mathbb{R}^4\). The corresponding gauge field (curvature) is the \(g\)-valued 2-form \(F := DA = dA + [A, A]\) where \(DA\) is the covariant exterior derivative of \(A\). Two gauge potentials \(A\) and \(A'\) are regarded as being equivalent if they are related by a gauge transformation \(A' = A^* \Omega + \Omega^{-1} dA\), where \(\Omega\) is a \(G\)-valued function on \(\mathbb{R}^4\).

The self-duality equations are \(*F = F\) where \(*\) is the Hodge duality operator. These equations form a set of coupled first-order nonlinear partial differential equations for \(A_i\). They are underdetermined (fewer equations than unknowns), but this underdeterminacy can be removed by imposing a 'gauge condition' such as \(A_0 = 0\). The self-duality equations are invariant under gauge transformations. The self-duality equations are completely solvable as a consequence of the twistor correspondence. The equations possess the Painlevé property. Many other integrable partial differential equations which also have the Painlevé property can be derived from the self-dual Yang Mills equation.

Finally, let us discuss an interesting connection of the self-dual Yang Mills equation and the Yang Mills equation. From the self-dual Yang Mills equation we find, after reduction, the system of partial differential equations\(du_1/dt = u_3 v_3, du_2/dt = u_1 v_3, du_3/dt = u_1 u_2\) When we differentiate this system with respect to \(t\) and insert it into the new second order equations we arrive at
\[
\frac{d^2 u_1}{dt^2} = u_1(u_3^2 + u_2^2), \quad \frac{d^2 u_2}{dt^2} = u_3(u_1^2 + u_2^2), \quad \frac{d^2 u_3}{dt^2} = u_3(u_1^2 + u_2^2).
\]

This system does not have the Painlevé property. It is not integrable. From the Yang-Mills equation \(D(F) = 0\) we find, after reduction, the nonintegrable system
\[
\frac{d^2 u_1}{dt^2} = -u_1(u_3^2 + u_2^2), \quad \frac{d^2 u_2}{dt^2} = -u_3(u_1^2 + u_2^2), \quad \frac{d^2 u_3}{dt^2} = -u_3(u_1^2 + u_2^2).
\]

This system is nonintegrable and can be derived from the Hamilton function \(H(u, \dot{u}) = (u_1^2 + u_2^2 + u_3^2)/2 + (u_1^2 u_3^2 + u_2^2 u_3^2 + u_3^2 u_1^2 + u_3^2 u_2^2)/2\). It shows chaotic behaviour. Furthermore, we find that it does not pass the Painlevé test. The resonances are given by \(-1, 1\) (twofold), \(2\) (twofold) and 4. Studying the behaviour at the resonances we find a logarithmic psi-series. The two systems are equivalent up to the sign on the right hand side. Since we consider the system in the complex domain for the Painlevé test we can find that the two systems are related via the transformation \(t \rightarrow it\).

**REFERENCES**

Nonlinear dynamical systems often have more than one attractor. The basin boundary is the set of all points for which each open neighborhood contains points of at least two different domains of attraction. A generalized attractor A is the union of finitely many attractors, and we define basin(A) to be the interior of the closure of the domain of attraction of A. We now assume for the transient region R that there exist 2 generalized attractors A and B, and each point in R that escapes from R under iteration of the map F is either in basin(A) or in basin(B). The stable set S(R) might contain points on the basin boundary that are accessible from basin(A) (or basin(B)) but not accessible from the transient set R\S(R). Hence, the ASST method for finding accessible points on S(R) is, generally speaking, not a procedure for finding accessible points on the basin boundary. A third question is: "Given a line segment J that has one end point in basin(A) and the other end point in basin(B). Describe a procedure for finding a numerical trajectory on the basin boundary that starts on J and which is accessible from basin(A)."

The numerical methods ("straddle methods") described below have been developed in collaboration with J.A. Yorke. We restrict our attention to cases in which there is only one positive Lyapunov exponent.

For the straddle methods below, we use the notions of "escape time" and "ε-refinement". The escape time T(x) of a point x in the transient region R under the map F is the minimum value n such that F^n(x) is not in R; the escape time T(x) is infinity if F^n(x) is in R for all n. For (x,y) on a line segment J we always assume for convenience that the ordering on J is such that we may write x < y; we denote [x,y] for the segment joining x and y, and |x-y| for the distance of x and y. A second question is: "Given a line segment J that crosses S(R) transversely. Describe a procedure for finding a numerical trajectory on the stable set S(R) that starts on J and which is accessible from transient set R\S(R)."

A point p in S(R) is accessible from an open set V if there is a continuous curve K ending at p such that K\(\setminus\{p\}\) is in V. We first investigate the case where V is the transient set R\S(R). A second question is: "Given a line segment J that crosses S(R) transversely. Describe a procedure for finding a numerical trajectory on the stable set S(R) that starts on J and which is accessible from transient set R\S(R)."

STRADDLE METHODS. Straddle methods involve a refinement procedure in which 2 points on a curve segment are replaced by 2 new points. In some cases the points have different roles. Usually each of the refinement procedures takes a pair of points and returns a pair of points; such a returned pair is on the line segment joining the two points of the original pair. The distance between the two points in the returned pair is smaller than the distance between the points of the original pair. Straddle methods consist of applying the refinement procedure repeatedly until the points in the resulting pair are less than some specified distance σ apart, say σ = 10^{-5}. If the points in the original pair are already less than σ apart, then no refinement is carried out. Next apply the dynamics; that is, apply the map F to each of the 2 points of the resulting pair, giving a new pair.

The basic numerical method takes a pair \(\{a_n,b_n\}\) which is separated by at most a distance σ, and applies the map F to each of the points of this pair. If the new pair \(\{F(a_n),F(b_n)\}\) is separated by less than σ,
then it is denoted \( (a_{n+1}, b_{n+1}) \), and otherwise the refinement procedure is applied repeatedly until a pair with separation at most \( \sigma \) is obtained, and it is called \( (a_{n+1}, b_{n+1}) \). However, in order to produce the first pair \( (a_0, b_0) \), the method starts by applying the refinement procedure on the given pair \( (a, b) \), whose points are presumably more than \( \sigma \) apart. Writing \( I_n \) or \( [a_n, b_n] \) for the line segment from \( a_n \) to \( b_n \), and to the precision of the computer we usually have \( I_n \subset F(I_n) \). We call the sequence of tiny straight line segments \( \{I_n\}_{n \geq 0} \) a straddle trajectory.

**SST METHOD.** The "saddle dynamic restraint problem" is to describe a numerical method for finding a trajectory that remains in a specified transient region for an arbitrarily long period of time. First, we describe the refinement procedure that is involved in the current straddle method. Let \( (a, b) \) be a pair such that \( [a, b] \) intersects \( \overline{S(R)} \) transversally.

Let \( (\alpha, \gamma, \beta) \) be a triple on \([a, b]\). We call \( (\alpha, \gamma, \beta) \) an Interior Maximum triple if both \( T_R(\gamma) > T_R(\alpha) \) and \( T_R(\gamma) > T_R(\beta) \); we call \( (\alpha, \gamma, \beta) \) a PIM triple if \( (\alpha, \gamma, \beta) \) is an Interior Maximum triple and \( |\beta-\alpha| < |b-a| \). Assume that \( (\alpha, \gamma, \beta) \) is an Interior Maximum triple, and let \( P \) be any \( \sigma \)-refinement of \((\alpha, \beta)\) including \( \gamma \). The procedure that selects in the refinement \( P \) any PIM triple \( (\alpha, \gamma, \beta) \) is called a PIM triple (refinement) procedure.

The solution to the "saddle dynamic restraint problem" is the straddle trajectory using the PIM triple procedure. We call the sequence of tiny straight line segments \( \{I_n\}_{n \geq 0} \) a saddle straddle trajectory or SST trajectory, and we call the straddle method that generates the SST trajectory \( \{I_n\}_{n \geq 0} \), the SST method. Notice that each tiny line segment in an SST trajectory straddles a piece of \( (\text{chaotic}) \) saddle. An SST trajectory typically resembles (after a few iterates) a basic set in the (chaotic) saddle.

**ABST METHOD.** The "accessible saddle dynamic restraint problem" is to describe a numerical method for finding a trajectory on the stable set \( S(R) \) that is accessible from the transient set \( R \setminus S(R) \). The refinement procedure that is involved in the current straddle method is a PIM triple (refinement) procedure in which a PIM triple \( (\alpha, \gamma, \beta) \) is selected from the \( \sigma \)-refinement \( P \) of the Interior Maximum triple \( (\alpha, \gamma, \beta) \) such that \( \{a, a^*\} \) is in the transient set. This refinement procedure is called the Accessible PIM triple (refinement) procedure. The solution to the "accessible saddle dynamic restraint problem" is the straddle trajectory using the Accessible PIM triple procedure. We call the straddle trajectory \( \{I_n\}_{n \geq 0} \) an accessible saddle straddle trajectory or ABST trajectory, and we call the straddle method that generates the ABST trajectory \( \{I_n\}_{n \geq 0} \), the ABST method. An ABST trajectory typically resembles (after a few iterates) a subset of the nonwandering points in \( R \) which are accessible from the set \( R \setminus S(R) \).

**REFERENCES**


Abstract

The present discussion attempts to show some mathematical connections between chaotic dynamics, fractal sets and dimensionality which may have relevance to physical systems.

1. Introduction

At least since the space time of relativity four dimensions have occupied a special place in physics. Modern nonlinear science has reinforced this situation. O. Roessler for instance has repeatedly drawn attention to four dimensional sets as the frontiers of new phenomena associated with wrinkled and hairy attractors [3]. Puelle, Takens and Newhouse envisages chaos as a sequence of a finite number of Hopf bifurcations leading to a totally unstable torus in four dimensions [4]. In fact one of the most important discoveries in topology implies that four dimensions are more complicated than any lower dimension and surprisingly even higher dimension [5]. To motivate our approach consider a recent result [2] where it was found that haemoglobin has a surface fractal dimension \( d_s = 2.4 \). Now this is more profound than finding for instance a two dimensional object with a fractal dimension less than two because somehow holes could account for the reduction. Here however they are insisting that it is a surface. Consequently the 0.4 may be attributed to "negative" holes in the surface. It must be a very rugged surface full with little mountains trying in a sense to jump out into the surrounding three dimensions. Should we ever encounter a surface which has a fractal dimension \( d > 3 \), this would be even more radical. In a "facon de parler" such a surface wants to reach out of the embedding three dimensional Euclidean manifold. That way \( d = 3 \) may be regarded as a "critical" dimension for this "strange" surface. Generalizing to \( n \) dimensions we could say that whenever the fractal dimension of an object and the dimension of the hosting manifold becomes equal a critical state in the preceding sense is reached. For reasons which we are about to make clear here, we will term this critical state quasi ergodic criticality. In what follows we would like to show that under a fairly reasonable assumption one can conclude that four dimensions marks a special point in chaotic dynamics.

2. Cantor-like sets and critical ergodicity

The starting point of our analysis is the generally accepted realization that fractals [6] are the carriers of complex strange behaviour. Second and without going into detail, we follow Yorke's conjecture that single Cantor sets are some how the back bone of all strange behaviour [7]. To that we add what we intuitively feel as evident namely that in one dimension it is extremely hard to think of any splier fractal than Cantor's middle third set [2] with \( d_c = \log 2/\log 3 \). If we can accept all this, which is not particularly easy to justify we can claim that in four dimensional phase space a strange set will typically have a Cantor-like fractal dimension \( d_c \neq 4 \). This result is reached using the following scaling argument. The idea is to find the equivalent to a triadic Cantor set in two dimensions. Such a set should be triadic Cantorian in every conceivable direction. It cannot therefore be the Cartesian product of two such sets, \( d = \log 4/\log 2 \) nor a Cantor target \( d_c = 1 + \log 2/\log 3 \). However we know that a unit area \( A \) of an Euclidian manifold is given by \( A = (1)(1) = 1 \) and consequently a corresponding or quasi area of a Cantor set is \( A_C = (d_c) (d_c) \). It follows then that in order to normalize \( A_C \) it must be multiplied by the normalization factor \( S_2 = (A/A_C)_1 \). By analogy in \( n \) dimensions we would have \( S_n = (A/A_C)_n \). Denoting the \( n \)-th Cantor-like fractal dimension in \( n \) dimensional space by \( d_C(n) \) and the dimension of the corresponding Euclidean space in \( n \) dimensions by \( d_s = n \) it follows then that

\[
\left( \frac{d_s}{d_c} \right)^n = S_n = \left( \frac{1}{d_c} \right) \left( \frac{d_c}{d_s} \right)^n = \left( \frac{d_c}{d_s} \right)^n
\]

where \( d_s \) is termed the escalation factor. This is the set which we are looking for and the result is now evaluated for \( d = \log 2/\log 3 \) in table 1. Note that \( d_s \) could be equally interpreted as the Froligt multiplier of a discrete map

\[
d_{(m+1)} = \frac{d_m}{d_s}
\]

where \( n = m + 1, \quad d^{(0)} = d_c \) and \( d_s = \partial(d_{(m+1)})/\partial(d_{(m)}) \)

<table>
<thead>
<tr>
<th>( n )</th>
<th>( d_s )</th>
<th>( d_C )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1.58496</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2.51210</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>3.98159</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>6.31067</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>10.00218</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>15.85309</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>25.12655</td>
<td>8</td>
</tr>
</tbody>
</table>

TABLE 1 - There are a few interesting observations here. First \( d_C(n) \) is the fractal dimension of the Serpenski gasket which is the prototype of fractal lattices with infinite hierarchy of loops. Second for all \( n < 4 \) we have \( n > d_C \), while for \( n > 4 \) we have \( d_C > n \). Only at \( n = 4 \) we have a Cantor-like structure which comes very near to a space filling set. The two dimensional geometrical analogue of this is the peano curve which is ergodic and shares a few properties with fat fractals [8]. We may say therefore that at \( n = 4 \) the set is almost
This is strongly reminiscent of the Fibonacci role of the proximity of some rational and of a Feigenbaum cascade. Note also that Log 2/Log 6 is clearly a reasonable measure of i variable phase space x nonlinearly oscillating set described by a dimensional phase space is required to study speculate if fully developed turbulence has a value found frequently in two dimensional logistic map as well as the Stuttgart (1989). This is however another way of viewing the Chaos. McGraw Hill, London (1990).

Looking back at table 1. One may be lead to speculate if fully developed turbulence has a fractal dimension d \geq 6.3 and that five dimensional phase space is required to study this phenomena. This would be for instance a nonlinearly oscillating set described by a phase space \( x \) and \( \beta \) representing the temporal and spatial oscillation of a state variable \( x \). In addition we need a special fluctuation \( \omega \) and a temporal fluctuation \( \omega \) as forcing frequencies. This makes them indeed five variables. Another worthwhile observation is that the Fibonacci fractal dimension

\[ d_{F} = \frac{1}{(Log 2/Log 3)} = 2.58496 \]

is identical to \( d_{c} = \frac{1}{(Log 2/Log 6)} \) where Log 2/Log 6 is clearly a reasonable measure of the fractal dimension at period 3 chaos of a Feigenbaum cascade. Note also that \( d_{c} = Log 2/Log 6 \times 0.387 \) is very close to the smallest value found for period 3 chaos of the logistic map [9,11] (d \times 0.378).

It is, of course important to appreciate the role of the proximity of some rational and irrational numbers such as 3/4, W/5, and \( \phi = (1 + 5)/2 \) and \( d_{c} = Log 2/Log 3 \), in arriving at some of the preceding conclusions.

References
ABSTRACT

Typical similarities and differences between strange chaotic and nonchaotic attractors in deterministic systems and random behaviour are discussed. It has been shown that based on a single time series it is impossible to distinguish between these types of behaviour even using Lyapunov exponents technique.

In last decade attention has been given to a class of dissipative dynamical systems that typically exhibit strange behaviour [1-3]. Such behaviour has been found in numerical experiments [3,4] as well as in experimental systems [5,6].

Recently two classes of strange attractors have been distinguished:

(a) a strange chaotic attractor - one which is geometrically 'strange' i.e. the attractor is neither a finite set of points nor it is piecewise differentiable and one for which typical orbits have positive Lyapunov exponents

(b) a strange nonchaotic attractor - one which is also geometrically 'strange' but for which typical nearby orbits do not diverge exponentially with time (7-9, 15-21).

Strange nonchaotic attractors have been found to be typical for quasiperiodically forced systems [7,18]. Recently they have been also observed in experimental system [21].

Although one may doubt that these are periodic or quasiperiodic orbits with sufficiently long period but even in this case period is longer than any reasonable observation and that is why their name is justified.

As both types of strange attractors look very similar (compare for example Poincaré maps of Figure 1(a) and (b)), the value of Lyapunov exponents seems to be the only quality which allows us to distinguish these classes.

In this paper we present some numerical experiments showing that it is impossible to distinguish between strange chaotic and nonchaotic deterministic behaviour basing on a single time series.

For systems which equations of motion are explicitly known and the linearized equations exist there is a straightforward technique [10,11] for computing a complete Lyapunov spectrum.

For most of the experimental systems the equations of motion are not known [12] or are in the form for which the linearized equations do not exist [6,13]. In this case Lyapunov exponents are estimated based on the monitored long - term time series. First the attractor is reconstructed by the well-known technique with delay coordinates [14]. Our reconstructed attractor though defined by a single trajectory can provide points that may be considered to lie on different trajectories. It has been shown that in many cases this attractor has got Lyapunov spectrum identical to that of the original attractor [10].

The technique of estimation of Lyapunov exponents based on the reconstructed attractor gives good results when we have at least one positive Lyapunov exponent in the spectrum.

In what follows we consider the quasiperiodically forced Van-der-Pol's equation

\[ \ddot{x} - a(1 - x^2)x + x = \cos(2\pi ft) \]  

(1)

For eq. (1) linearized equations exist and we can compute Lyapunov exponents directly from the formula:

\[ \lambda = \frac{1}{\Delta t} \frac{d(t)}{d(0)} \]  

(2)

where: \( d = \sqrt{y^2 + \dot{y}^2} \), while \( y \) is a solution of a linearized equation.

In these cases we have computed Lyapunov exponents twice from the formula (2) and from time series based on Wolf et al. algorithm [10]. In numerical simulations the four order Runge-Kutta method with time step \( T/200 \), where \( T=2\pi/f \) has been used. Strange nonchaotic attractors have been observed up to \( T=10^5 \). The comparison of these results is shown in Figure 2.

From Figure 2 one finds that the calculation of the Lyapunov exponents from the explicitly known differentiable equation allows to distinguish between strange chaotic and nonchaotic attractors. This distinction cannot
be followed based on the Lyapunov exponents estimated from a single time series, as by this method we obtained positive values of Lyapunov exponents not only in the case of strange chaotic attractors but for strange nonchaotic attractors as well. This result may look quite surprising but it is justified when we follow the method of Lyapunov exponents estimation from the attractor reconstructed from single time series. If a time series is irregular (not periodic, quasiperiodic) it is not distinctive from chaotic one and the reconstructed attractor has got the complicated geometry. Estimating Lyapunov exponents from this attractor we have to obtain a positive values for both strange chaotic and nonchaotic attractors, as the whole procedure explores the aperiodicity of time series and not the explicit dependence on initial conditions. As other methods of estimating Lyapunov exponents from time series (23-26) are also based on the same method of attractor reconstruction it seems that using them similar results are very likely. The possibility of having a system showing strange behaviour without sensitive dependence on initial conditions should not be overlooked. It seems that more care will have to be given in applying the procedure of estimation of Lyapunov exponents from time series to experimental data. The general conclusion that it imply can be misleading, as there are systems for which distinction between strange chaotic, strange nonchaotic behaviour is impossible based on a single time series.

REFERENCES

INFLUENCE OF FRICTION ON THE CHAOTIC DYNAMICS IN COUPLED OSCILLATORS.

JAN AWREJCEWICZ AND WOLF D. REINHARDT

The University of Tokyo
Department of Mechanical Engineering
7-3-1 Hongo, Bunkyo-ku,
Tokyo, JAPAN

The University of Tokyo
Technical University
Department of Mechanical Engineering
Langer Kamp 19B,
3300 Braunschweig, GERMANY

Abstract - A route to chaos in the system with dry friction is analyzed. In spite of the complexity of the system, a similar transition to that discovered in the two-well potential anharmonic oscillator is described and illustrated. Dry friction weakens the chaotic dynamics and induces the occurrence of stick and slip transitions during the chaotic wandering of the trajectory in four dimensional phase space.

1. INTRODUCTION

The aim of this paper is to show the "qualitative universal" transition to chaos in a certain subclass of sinusoidally-driven nonlinear oscillators, i.e., systems with a two-well potential. The question of interest is whether or not the scenarios leading to chaotic orbits discovered in simple uncoupled oscillators are likely also hold for much more complicated systems, such as coupled nonlinear sinusoidally driven oscillators. Simulation experiments show that the potential has two wells, and that chaotic dynamics will obtain, in which each of the oscillators jumps between two wells in an unpredictable way.

2. THE SYSTEM

We consider a system of two coupled mechanical oscillators, both of which are externally driven. The governing equations are:

\[ m_1 \ddot{x}_1 + (C_1 - C_3)x_1 - C_3x_2 + C_2x_1^2x_1 + (k_1 + k_3)x_1 \]
\[ -k_2x_2 + k_3x_1^2 + \mu m_1 g \sin(\alpha) = q_1 \cos(\omega_1 t + \phi), \]
\[ m_2 \ddot{x}_2 + (C_3 - C_4)x_2 - C_4x_1 + C_6x_2^2x_1 + (k_3 + k_4)x_2 \]
\[ -k_3x_1 + k_4x_2^2 = q_2 \cos(\omega_2 t), \]

where \( m_1 \) and \( m_2 \) are the masses of the oscillators, \( C_1 - C_3 \) and \( k_1 - k_3 \) are damping and stiffness coefficients, respectively, \( q_1 \) and \( q_2 \) are the amplitudes of the exciting forces with corresponding frequencies \( \omega_1 \) and \( \omega_2 \), and \( \phi \) denotes a phase shift between the exciting forces.

In non-dimensional form we have:

\[ \xi_1'' + (\alpha_2 - \alpha_1)\xi_1' - \alpha_2(KM^{-1})^{0.5}\xi_2' + \gamma_1, \xi_1' - (x_1' \times x_3)\xi_1' = 0, \]
\[ -k_3(KM^{-1})^{0.5}\xi_2' + R \tan(\xi_1') = B_1 \cos(\tau + \phi), \]
\[ \xi_2'' + M(\alpha_2 - \alpha_3)\xi_2' - M \xi_2' + \gamma_2, R \xi_2'' = M(k_3 + k_4), \xi_2', \]
\[ -Mk_2(MK^{-1})^{0.5}\xi_1' - \xi_2'' = M^{0.5} K^{-0.5} B_2 \cos(\nu t). \]

where

\[ \tau = \omega_1 t, \xi_1' = (\omega_1 m_1^{0.5})^{-1} k_2^{0.5} x_1, \xi_2' = (\omega_2 m_2^{0.5})^{-1} k_2^{0.5} x_2, \]
\[ M = m_1 m_2^{-1}, K = k_3 k_4^{-1}, \nu = \omega_2 \omega_1. \]

Using the transformations (3), the nineteen parameters of equations (1) are reduced to fourteen parameters in (2).

Such a general system has been investigated earlier by the authors [1-3] using a systematic numerical approach. Transitions between quasiperiodic, strange chaotic and strange non-chaotic attractors have been reported as well as some special chaotic dynamics has been discussed and illustrated. Here an attention is focused on the influence of friction of the chaotic dynamics on the mentioned above system.

3. NUMERICAL ANALYSIS

We define:

\[ F = (k_1 + k_3)x_1 - k_3 x_2 + k_2 x_2^2 - C x_2 - q_1 \cos(\omega_1 t). \]

When \( \xi_1 = 0 \) and \( |F| < |F_{\text{th}}| \), the first oscillator is in a stick state. During the transition from a slip-state to a stick-state, an acceleration jump occurs. During an exit from a stick-state to a slip-state the acceleration is continuous, but a jump in the third derivative of the displacement appears. The velocity in both cases is always continuous. During the "transition over a stick area", but without sticking, the acceleration jump also occurs. When a regular transition, however, from a slip-state to a stick takes place with \( \nu = |(F)|_{\text{max}} \), the one eigenvalue of the Jacobi matrix is equal to zero. When \( |F| = |(F)|_{\text{max}} \) and the acceleration jump does not appear, the Jacobi matrix is not singular. The following equations govern the dynamics in the stick-state:

\[ \xi_1 = 0, \]
\[ \xi_2'' + M(\alpha_2 - \alpha_1)\xi_2' - \gamma_2 R \xi_2'' = M(k_3 + k_4) \xi_2' - M^{0.5} K^{-0.5} B_2 \cos(\nu t). \]

with the following transition-condition:
We consider the behavior of the system (2) for the following fixed parameters:

\[ y = M = K = 1.0, \quad \kappa_1 = \kappa_4 = 0.816326, \quad \phi = 0.0, \]

\[ \gamma_1 = \gamma_2 = 0.3, \quad B_1 = 0.05, \quad B_2 = 0.2, \]

and for two values of friction \( R \).

Example 1 (\( R = 0.05 \)).

We take \( \alpha_1 = \alpha_4 = 0.01 \) and \( \alpha_3 = \kappa_3 = 0.3 \). For these parameters and without friction (\( R = 0 \)), the system exhibits intermittent chaos. Friction dampens the chaotic dynamics of the orbits and for \( R = 0.05 \) we find a quasiperiodic attractor. In the neighborhood of these parameters (for \( \alpha_1 = \alpha_4 = 0.05 \) and \( \alpha_3 = \kappa_3 = 0.3 \)), a periodic attractor is found. An increase in \( \alpha_1 \) \((\alpha_1 = \alpha_4)\) results in an increase in the magnitude of the self-excited oscillations. The periodic orbit grows and finally leads to intersections of the stable and unstable manifolds and a trajectory starts to wander in an unpredictable way between two potential wells. This situation is illustrated for \( \alpha_1 = \alpha_4 = 0.2 \) in Fig.1.

Example 2 (\( R = 0.1 \)).

In the second example we analyze the influence of the coupling between two oscillators. The numerical calculations have been carried out for the same parameters as in Example 1 and additionally for \( R = 0.1 \) and \( \alpha_1 = \alpha_4 = 0.2 \). When two oscillators are strongly coupled \((\alpha_3 = 2.0, \quad \kappa_3 = 0.3)\), a periodic orbit is found. This orbit lies to the right of the origin. However, to the left of the origin there is also another small periodic orbit. These two orbits lie in two isolated potential wells. Decreasing \( \kappa_3 \) causes the trajectory to move from the potential well and start to wander between the two potential wells (Fig.2). The escape, however, from one of the wells to the other is rather rare. The possibility of it occurring increases with a further decrease in \( \alpha_1 \). For example, for \( \alpha_1 = 0.3 \), one of the projections of the Poincaré map shows a very complicated dynamics (Fig.3).

In order to understand how two oscillators move in a chaotic manner, two time histories (for relatively long time intervals of the same chaotic attractors) are presented in Fig.4. In this figure one can also observe stick states. These states correspond to the very short horizontal parts of \( \xi_1(t) \).

4. CONCLUSIONS

In the six-dimensional nonlinear mechanical system with friction that was investigated, quasiperiodic and chaotic attractors are detected. We have discussed and illustrated that in this case the route to chaos is the same as in the simple two-well potential, sinusoidally-driven oscillator. An investigation of the influence of dry friction on the chaotic behavior of two coupled oscillators shows that increasing the friction weakens the chaotic dynamics of the orbits. During the chaotic motion of the first oscillator, stick-slip transitions are observed.

REFERENCES


"Indirect" Time Series Analysis for One-Dimensional Chaos Based on Perron-Frobenius Operator

Tohru KOHDA† and Kenji MURAO††

Kyushu University*† and Miyazaki University††
Fukuoka 812† and Miyazaki 889-12††, Japan

Abstract A unified approach to time series analysis for one-dimensional discrete chaos is given which is based on the Galerkin approximation to the Perron-Frobenius integral operator. The proposed method gives approximations with high accuracy to statistics of various chaos. Numerical results for 1/f² power spectrum of intermittent chaos also show that in the limit of zero frequencies, the observed exponent δ of the FFT power spectrum of long-time trajectories is in good agreement not with the Procaccia-Schuster's estimate but with ours.

I. Introduction

There are two kinds of time series analysis for long-time chaotic time series analysis. One is ensemble average technique, in which we evaluate certain statistics of a sample time-long trajectory {xₙ}ₙ=₀, with some initial value x₀ = x₀; the other one is the "ensemble-average technique" under the assumption that x is mixing with respect to an absolutely continuous invariant measure, denoted by f*(x)dx. We give a unified approach to time series analysis for chaos by such an ensemble-average technique.

The time-average technique which is a usual method [2] is referred to as the "direct method". On the contrary, the ensemble average technique is a kind of "indirect methods" because there is no need to directly calculate trajectories. Hence such an indirect method is expected to play an important role in theoretically understanding chaos. In fact, the existence of the Perron-Frobenius operator, denoted by P, [3][4] is important. This operator, however, gives no practically calculating method because of its infinite dimensionality. Such a situation leads us to consider an efficient algorithm for systematically calculating statistics which is based on the Galerkin approximation to P, on a suitable function space. Numerical experiments demonstrate that the proposed method can give approximations with high accuracy to statistics of various chaos.

II. Perron-Frobenius Operator and Statistics of Chaos

If y = τ(z) is mixing with respect to f*(x)dx, then for almost initial value x = x₀ sequences {xₙ}ₙ=₀ can chaotically behave. From the Birkhoff individual ergodic theorem, the time average of any L₁ function F(x) along a trajectory {xₙ}ₙ=₀, which is defined by F = limₙ→∞ ½ ∑ₙ=₀ f(τ(z)) is equal almost everywhere to the ensemble average of F(x) over f, defined by < F >= ∫ f P₁ f*(x)dx. The direct time series analysis is based on using F. However, the dependence on initial conditions, one of chaotic properties[1], prevents us from precisely evaluating F. On the other hand, the indirect time series analysis is based on using < F >. We begin with reviewing relations between typical statistics and F₁.

The operator P₁ is defined by P₁h(x) = ∫ f(τ(z)-y)h(z)dy. For any f functions of bounded variations g(z) and h(z), P₁ has the important property (fₙ(z), h(τ(z))) = (P₁g₁(z), h(z)), where (g₁, h₁) = ∫ f₁ g(z)h₁(z)dx. The invariant density f*(z) which plays a key role in our indirect method is the eigenfunction of P₁ belonging to the eigenvalue 1, that is, P₁ f*(z) = f*(z). The autocorrelation function is defined by ρ(k) = < z² > = < z² >² - < z >². The first term of the rhs of this equation is rewritten as < z² >² = (P₁ f*(z), z), where the above property of P₁ is repeatedly used. Let h₁(z) be the eigenfunction of P₁ with the eigenvalue λ₁ for the eigenvalue problem P₁ h₁(z) = λ₁ h₁(z) [2]. If we can expand f*(z) as f*(z) = ∑ₙ=₀ h₁(z), then we have ρ(k) = ∑ₙ=₀ uₙ², the Fourier Transform of which gives the power spectrum S(ν) = ∑ₙ=₀ uₙ²/(1 - λ₁)(1 - λ₁, z⁻¹)

where λ₁ = 1, uₙ = hₙ(z), and z = exp(i2πν) with 0 < ν < 1. Oono and Takahashi[5][6] demonstrated that the Fredholm theory of P₁ plays an important role in discussions of the power spectrum. It is, however, difficult to find exact solutions of eigenvalues and eigenfunctions of P₁, because P₁ has the limit of infinite dimensionality. Such a situation led us to consider an efficient algorithm of the indirect method.

III. Galerkin Approximations to the Perron-Frobenius Operator

Let Δ be a function space which is spanned by a vector basis function ϑ(z). Each component of ϑ(z), denoted by eₙ(z), is an appropriately chosen piecewise polynomial of at most D degree whose combination approximates f*(z) by the Galerkin method [5] such as f*(z) = ∑ₙ=₀ eₙ(z), where the superscript t denotes the transpose of the vector. Using ϑ(z), we get < z² > = ∑ₙ=₀ eₙ(z), eₙ(z). Furthermore, using the Galerkin method with ϑ(z) on Δ, we approximate P₁ ϑ(z) such as P₁ ϑ(z) ≃ ∑ₙ=₀ eₙ(z), which leads us to readily obtain < z² > = ∑ₙ=₀ eₙ(z), eₙ(z), where h(D+1) x N(D+1) matrix P₁ is referred to as the Galerkin-approximated matrix of the Perron-Frobenius operator where N and D are integers to be given below. The explicit form of P₁ is given in[2]. Let h₁ be the i-th right eigenvector of P₁ with the eigenvalue λ₁ for the easily tractable eigenvalue problem P₁ h₁ = λ₁ h₁.

The constructing method of Δ is as follows. We divide f into N subintervals [Iₙ] with partition points {cₙ}ₙ=₀ satisfying 0 = c₀ < c₁ < c₂ < ··· < cₙ = 1 such that f = ∑ₙ=₀ eₙ(z), eₙ(z) = ∑ₙ=₀ eₙ(z), eₙ(z). The above Galerkin approximations depend on the appropriate selections of {cₙ}ₙ=₀ and of ϑ(z) [5]. A simple but efficient procedure, however, is omitted here for selecting {cₙ}ₙ=₀. Next, we take bases eₙ(z) such as eₙ(z) = pₙ(z)(z), cₙ(z), 0 ≤ k ≤ D, 1 ≤ n ≤ N. In the above equation, cₙ(z) is the characteristic function of Iₙ and pₙ(z) is the k-th order Legendre's polynomial which is orthogonal to each other on Iₙ. For most of practical usages, we use D = 2. When t has a bounded invariant density, the function S(ν), referred to as a supplementary function, is taken to be 1. On the other hand, t has an unbounded invariant

857
density, \( s(x) \) is chosen to be a singular function which approximates to singularities of the unbounded invariant density and the inner product \((g, h)\) must be also replaced by the weighted inner product \((g, h)w = \int g(x)h(x)w(x)dx \) with the weighting function \( w(x) = s^2(x) \).

IV. Numerical Examples

Example 1

Let

\[
\tau(x) = \begin{cases} 
ax^2 + (a + b - ab)/b & 0 \leq x \leq x_p = (1 - 1/b)^{1/t} \\
-b(x^2 - 1) & x_p < x \leq 1 \end{cases}
\]

This map can generate periodic chaos for suitable parameters. Figures 1 and 2 show \( \tau(x) \) and the power spectrum \( \tilde{S}_T(\nu) \) for periodic chaos of period 6 which are calculated by our method. In this calculation, we take \( \{r^n(0)\}_{n=1}^{100} \) as the partition points \( \{e_i\}_{i=1}^{N-1} \) so that edges of the support of \( \tau(x) \) will coincide with the partition points. In the calculation of \( \tilde{S}_T(\nu) \), the finite discrete Fourier transform of \( \{\rho(k)\}_{k=0}^{T-1} \) \((T = 1024 \times 6)\) is used instead of using (1). On the other hand, \( \tilde{S}_{T,m}(\nu) \) is obtained by averaging \( m = 200 \) discrete Fourier transforms of trajectories of length \( T \). The spectrum \( \tilde{S}_T(\nu) \) is in good agreement with \( \tilde{S}_{T,m}(\nu) \) except for fluctuations in the latter.

Example 2

Let

\[
\tau(x) = \begin{cases} 
x + ux^2 & 0 \leq x \leq x_p \\
(x - x_p)/(1 - x_p) & x_p < x \leq 1 \end{cases}
\]

where \( \tau(x_p) = 1, u > 0, 1 < z < 2 \). This map generates intermittent chaos with the power spectrum \( 1/f^z \). Figure 3 shows the power spectrum \( S(\nu) \) by our method (the smooth solid line) and \( S_{T,m}(\nu) \) with \( T = 2^{16} \) and \( m = 100 \) by the direct method (the fluctuated line), each of which is in good agreement each other in wide frequency range. In applying our method, we used \( s(x) = x^{-(r-1)} \) because \( s \) has the unbounded invariant density with a \((z - 1)\) th order pole at \( x = 1 \). In this figure, the broken line shows the Procaccia and Schuster's estimate \( \hat{S}(\nu) \) of the spectrum when \( \nu \) goes to 0 which does not coincide well with the former two.

---

Fig. 1 Invariant density \( \tau(x) \) (by our indirect method) for periodic chaos of period 6 in the example 1.

Fig. 2 Power spectra \( \tilde{S}_T(\nu) \) (by our indirect method) and \( \tilde{S}_{T,m}(\nu) \) (by the direct method) for periodic chaos of period 6 in the example 1.

Fig. 3 Comparison of power spectra calculated by using three different methods for intermittent chaos in the example 2.

References

Constructive Implicit Function Theorem and its Application

Shin'ichi Oishi, Masahide Kashiwagi, Mitsunori Makino and Kazuo Horiuchi
School of Science and Engineering, Waseda University

Abstract
In this paper, a new algorithm of tracing solution curves for the homotopy method is presented. The algorithm is based on the predictor-corrector method and guaranteed that tracing solution curves always succeeds.

1. Introduction
Recently the study of the homotopy method have been made a great stride for solving nonlinear equations globally. In the homotopy method, for solving a nonlinear equation
\[ f(x) = 0, \quad f: \mathbb{R}^n \rightarrow \mathbb{R}^n, \] (1.1)
an auxiliary equation \( g(x) = 0 \) is used, having a trivial solution \( x_0 \in \mathbb{R}^n \). Changing the equation \( g(x) = 0 \) into \( f(x) = 0 \) gradually, a solution of \( f(x) = 0 \) is obtained by tracing change of the solution of (1.1). For the purpose, we introduce a homotopy equation with parameter \( \xi \)
\[ h(x, \xi) = 0, \quad h: \mathbb{R}^n \times [0, 1] \rightarrow \mathbb{R}^n, \] (1.2)
where
\[ h(x, 0) = g(x), \quad h(x, 1) = f(x). \] (1.3)
Typically, the following homotopy \( h \) is used:
\[ h(x, \xi) = (1-\xi)g(x) + \xi f(x). \] (1.4)
Then a solution \( x^* \) of \( f(x) = 0 \) can be obtained by tracing an implicitly defined solution curve \( h^{-1}(0) \) from \( (x_0, 0) \) to \( (x^*, 1) \).

For tracing such a solution curve, a kind of predictor-corrector method is known to be effective. In this method, a point on the solution curve is moved along the tangent vector of the curve (predictor), and correct an deviation of the moved point from the solution curve \( h^{-1}(0) \) (corrector) by, for example, the Newton method. However, the predictor-corrector method has a deficiency such that the method frequently fails in tracing solution curves since the Newton iteration is used in the method. Although theoretically we can avoid the failure of tracing if we choose a step length sufficiently small, it has been said that to estimate such a step length is impossible.

In this paper, under the assumption that the derivative of \( h \) is Lipschitz continuous, we shall present "constructive" implicit function theorem. By the theorem, we can estimate the radius of a neighborhood in which implicit function theorem is valid. Moreover, the theorem gives a numerical algorithm to calculate the implicit function in that neighborhood based on the simplified Newton method.

Moreover, using the theorem, we present an improved predictor-corrector algorithm of tracing solution curves. By the new algorithms, tracing solution curves of (1.2) is guaranteed to succeed.

2. Constructive Implicit Function Theorem
In the following, when a Banach space \( X \) is direct sum of a subspace \( X_1 \) and a subspace \( X_2 \), we write \( X = X_1 + X_2 \). Moreover, By \( B(c; \varepsilon; X) \) we denote a ball with center \( c \) and radius \( \varepsilon \) in \( X \).

The conventional implicit function theorem can be written as follows:

**Theorem 1** (Implicit Function Theorem)
Let \( X \) be a Banach space which is direct sum of subspaces \( X_1 \) and \( X_2 \), and \( Y \) a Banach space. \( U \subset X \) an open set, and \( g: U \rightarrow Y \) a \( C^1 \) operator. Assume that there exists a \( (p_1, p_2) \in X \) such that the following conditions hold:
\[ g(p_1, p_2) = 0, \]
and the partial derivative of \( g \) at \( (p_1, p_2) \) with respect to the second parameter is homeomorphism. Then for sufficiently small \( \varepsilon > 0 \) and \( \delta > 0 \), the following holds true:
An equation
\[ g(x_1, x_2) = 0, \quad x_2 \in B(p_2; \varepsilon; X_2) \] (2.1)
can be solved uniquely for \( x_1 \) with fixed \( x_2 \) \( \in B(p_1; \delta; X_1) \).

A map which maps \( x \) to \( x_2 \) is called an implicit function. In this theorem, one can not estimate the largeness of \( \varepsilon \) and \( \delta \). To estimate them, we assume that the Fréchet derivative of \( g \), \( Dg \), is \( \alpha \)-Lipschitz continuous. Then the implicit function theorem can be extended as follows:

**Theorem 2** (Constructive Implicit Function Theorem)
Let \( X \) be a Banach space which is direct sum of subspaces \( X_1 \) and \( X_2 \), and \( Y \) a Banach space. \( U \subset X \) an open set, and \( g: U \rightarrow Y \) a \( C^1 \) operator. Assume that Fréchet derivative of \( g \), \( Dg \), is \( \alpha \)-Lipschitz continuous. Moreover, we assume that \( \alpha \) \((\alpha, \beta) \in X \) and a bounded linear operator \( A: X \rightarrow Y \) are given and
that there exist \( r, d_1, d_2 \) such that the following conditions hold:

1. \( 1 \leq n \leq m \)
2. \( \| D(x, y) - A \| \leq k \)
3. \( A(x, y) = 0 \)
4. \( \| A(x, y) \| \leq 1 \)
5. \( \| A(x, y) \| \leq 1 \)

Then if \( \varepsilon > 0 \) and \( \delta > 0 \) satisfy

\[
\| A(x, y) \| \leq 1/\delta.
\]

(2.5)

Then 

\[
\| A(x, y) \| \leq 1/\delta.
\]

(2.6)

(2.7)

(2.8)

(2.9)

where \( A \) is an approximate matrix of \( Dh(x, y) \). Then \( X_t \) is determined to be \([q]\), a subspace which is generated by \( q \), and \( X_t \) is determined to be \([q]\), an orthogonal complement of \( X_t \). Here we note that in such decision, \( d_1 \) in Theorem 2 is estimated to be 0. Then \( A(0, \cdot)^{-1} \) exists if \( A \) is maximal rank.

We now present the algorithm for tracing solution curves.

**Algorithm 1**

Let \( X_t \) be an \( \mathbb{E}^n \), \( Y \) a \( \mathbb{E}^n \), \( h: \mathbb{E}^n \rightarrow \mathbb{E}^n \) a \( g \) operator. Assume that \( Dh \) is \( \alpha \)-Lipschitz continuous. Then we consider to trace solution curves of the equation \( h(x) = 0 \). Let \( x_0 \in X \) be a starting point on a solution curve, and \( q_1 \in X \) a vector representing a prediction direction.

1. Let \( i = 0 \).
2. Calculate \( h(x_0) \) and \( r = \| h(x_0) \| \).
3. Calculate an approximate matrix of \( Dh(x_0) \), and an error estimation \( K \) such that \( \| Dh(x_0) \| \leq K \).
4. Let \( q_1 \) be a solution of the equation such that \( q_1 \in N(A), \| q_1 \| = 1, q_1 \cdot q_i > 0 \).
5. Decide \( X_1 \) and \( Y \) to be \([q]\) and \([q]\).
6. Calculate \( \varepsilon > 0 \) and \( \delta > 0 \) such that Eq.(2.6) and Eq.(2.7) is satisfied. Here let \( d_1 = 0 \) and \( d_2 = 1/\| A(0, \cdot)^{-1} \| \).
7. Execute the iteration such that

\[
Y_{i+1} = y_i + q_{i+1},
\]

(3.7)

until \( \| h(y_{i+1}) \| \) becomes sufficiently small.
8. Let \( x_{i+1} = y_{i+1} \) and \( i = i+1 \). Go to 2.

**4. Concluding Remarks**

In this paper, assuming that the derivative of \( h \) is \( \alpha \)-Lipschitz continuous, we have presented constructive implicit function theorem which can estimate the radius of the neighborhood in which implicit function theorem is valid. Using the theorem, based on the predictor-corrector method, we have presented a new algorithm of tracing solution curve without failure in the homotopy method.

This paper is partially supported by the Grant-in-Aid for the Ministry of Education, Science, Research and Culture.

**References**

Abstract: The finite-difference, finite-element, and the algebraic multiconfiguration Hartree-Fock methods are briefly compared, and some advantages and disadvantages of the approaches are discussed.

1. INTRODUCTION

In traditional quantum chemistry and physics, the molecular or atomic orbitals (one-electron functions) are expressed as linear combinations of global basis functions of Slater (exp(-r^2)) or Gaussian (exp(-a^2r^2)) type. The truncation of the expansion leads to a basis-set truncation error (BSTE), the exact magnitude of which is difficult to establish. With the increase of the Hartree-Fock electron correlation, the BSTE has become a serious bottleneck in ab initio quantum chemistry. The BSTE can be systematically reduced to negligible magnitude by using numerical rather than algebraic approximation.

In the numerical approximation the unknown functions (orbitals) are expanded in piecewise differentiable local basis functions, while in the algebraic approximation global functions are used. In engineering sciences low-order polynomials can be used as local basis functions, while to achieve the high accuracy needed in quantum mechanical problems polynomials of fourth to eighth order are used. The numerical approach can be divided into at least two different classes: the finite-difference (FD) and the finite-element (FE) methods.

In this paper, the algebraic, the finite-difference, and the finite-element approaches in quantum mechanical problems will be compared, and some of the advantages and disadvantages of these methods will be discussed. In order to concentrate on the differences between the three approaches, we will restrict the discussion to atomic systems. For molecules, there are numerical methods for solving various local-density functional (LDF) equations using FD method [1,2], FE method [3,4], and splines in 2D [5] and in 3D [6] based on one-centre expansions. Fully numerical approaches to Hartree Fock equations for diatomic molecular orbitals are described here. For details, the reader is referred to the literature [15,16].

2. THE MCHF METHOD

For atoms, the orbitals are expressed in spherical coordinates (r,θ,φ), as products of a radial and an angular part

\[ \Phi = R(r) \times Y_{lm}(\theta, \phi) \]  

The angular parts \( Y_{lm}(\theta, \phi) \) are spherical harmonics, while the radial parts \( R(r) \) are expanded in local or global basis functions. Using the second-quantization formalism with normalized wave functions, the multiconfiguration Hartree-Fock energy is given as

\[ E = \sum_{\alpha \beta} \frac{\langle \alpha | \hbar \hat{r}_{ij} | \beta \rangle}{\langle \alpha | \hat{r}_{ij} | \beta \rangle} + \sum_{k \neq \alpha} \int \sum_{\alpha \beta} \frac{\langle \alpha | \hat{r}_{jk} | \beta \rangle}{\langle \alpha | \hat{r}_{jk} | \beta \rangle} \Phi_{\alpha jk} \Phi_{\beta jk} \]  

where \( \hat{r}_{ij} \) and \( \hat{r}_{jk} \) are the one- and two electron integrals, and \( \Phi_{\alpha jk} \) and \( \Phi_{\beta jk} \) are the elements of the one- and two electron density matrices, respectively.

\[ \hbar_{ij} = \int \Phi^*_\alpha(r) \Phi_{\beta}(r) \frac{1}{2} \frac{\partial^2}{\partial r^2} \Phi_{\beta}(r) \, dr \]  

\[ \delta_{\alpha jk} = \int \Phi^*_\alpha \Phi_{\beta} \Phi_{\gamma} \Phi_{\delta} \, d\tau_1 \, d\tau_2 \]  

\[ \Gamma_{ij} = \langle \alpha \beta | \Phi_{ij} | \alpha \beta \rangle \]  

\[ \Gamma_{jkl} = \langle \alpha \beta | \Phi_{jkl} | \alpha \beta \rangle \]  

\[ E = \sum_{\alpha \beta} \frac{\langle \alpha | \hbar \hat{r}_{ij} | \beta \rangle}{\langle \alpha | \hat{r}_{ij} | \beta \rangle} + \sum_{k \neq \alpha} \int \sum_{\alpha \beta} \frac{\langle \alpha | \hat{r}_{jk} | \beta \rangle}{\langle \alpha | \hat{r}_{jk} | \beta \rangle} \Phi_{\alpha jk} \Phi_{\beta jk} \]  

\[ \langle \Phi_{\alpha jk} | \hat{r}_{ij} | \Phi_{\beta jk} \rangle \]  

\[ \langle \Phi_{\alpha jk} | \hat{r}_{jk} | \Phi_{\beta jk} \rangle \]  

where \( \hbar_{ij} \) are the occupied orbitals, \( \hbar^2 \) in the kinetic energy operator, \( Z \) is the nuclear charge, \( r_{ij} \) is the interelectronic distance, \( \Phi \) is the wave function, and \( \delta_{ij} \) are the excitation operators [15].

In the MCHF method both the expansion coefficients of the orbitals (occurring in \( \hbar_{ij} \) and \( \hbar_{jk} \)) and the coefficients of the configuration state functions (occurring in \( \Gamma_{ij} \) and \( \Gamma_{jkl} \)) are optimized. In the Hartree-Fock method the number of configurations is restricted to one, and the density matrices are thus fixed.

3. THE FD-MCHF APPROACH

In the finite-difference (FD) approach, the Euler (Fock) equations are obtained from the energy functional (2), and the Lepicier is discretized using n-point formulas. In engineering sciences low-order (n=2-3) polynomials are mostly used, while to achieve the high accuracy needed in quantum chemical problems, one has to use high-order approximations (n=5 9/dimension). For each orbital i, the discretized coupled systems of non linear Fock equations

\[ \sum_{j \neq i} \int \sum_{\alpha \beta} \frac{\langle \alpha | \hat{r}_{ij} | \beta \rangle}{\langle \alpha | \hat{r}_{ij} | \beta \rangle} \Phi_{\alpha jk} \Phi_{\beta jk} \]  

are solved. The electron-electron interaction potentials \( V_{ij} \) are obtained using the Poisson equation

\[ \delta_{ij} = -\nabla \varphi \]  

The Lagrange multipliers of equation (7), \( \epsilon_{ij} \), which ensure the orthonormality of the orbitals, are calculated as expectation values. The equations (7) and (8) are solved until the residual vanishes and the energy becomes stationary. For simplicity, we above assumed the Hartree-Fock approximation. In the MCHF method, the configuration interaction (CI) coefficients are also optimized. This is done by constructing the Hamilton matrix of the chosen configuration space and diagonalizing it. After the diagonalization, the potentials which now include the configuration interaction are recalculated, and the orbitals are reoptimized. This cycle is repeated until the changes of the orbitals and those of the CI coefficients are negligible. A more detailed description of the FD-MCHF method is given in refs. [16,17].

4. THE ALGEBRAIC MCHF (MCSCF) APPROACH

In the algebraic approach, the orbitals of the energy functional (2) are expanded in global Slater or Gaussian functions. The energy function obtained is optimized with respect to the orbital parameters and the CI coefficients, with imposed orthonormality constraints. The orbital and the CI parameters appear in the integrals and the density matrices, respectively. The main features of the MCSCF optimization will be described here. For details, the reader is referred to the literature [15,16].

The variation of the MC energy function is described by two unitary operators \( \exp(S) \) and \( \exp(T) \) for orbital and configuration parameter variations, respectively. When the unitary operators are applied on a given state, the transformed wave function will remain normalized and the orbitals orthonormal. The \( S \) and \( T \) operators contain the independent variational parameters of the MC energy function.

\[ \langle \Phi | \hat{h} | \Phi \rangle = \sum_{i=1}^{n} \frac{1}{2} \frac{\partial^2}{\partial r^2} \Phi (r) \]  

\[ \Phi (r) = \sum_{n=1}^{N} \alpha_n \Phi_n (r) \]  

\[ \sum_{i=1}^{n} \frac{1}{2} \frac{\partial^2}{\partial r^2} \Phi (r) \]  

\[ \sum_{i=1}^{n} \frac{1}{2} \frac{\partial^2}{\partial r^2} \Phi (r) \]  

The Taylor series expansion of the energy function with respect to the parameters of the \( S \) and \( T \) can be constructed, and at the stationary point the first-order term (gradient) vanishes. The common choices for solving the optimization equations are the Newton-Raphson procedure or the Broyden update methods (quasi-Newton methods) [19,20]. The CI coefficients are usually calculated by using direct methods instead of explicit diagonalization of the Hamilton matrix.

5. THE FE-MCHF APPROACH

In the FE approach, the orbitals are expressed in local piecewise differentiable functions. We use Lagrange interpolation polynomials as basis functions [10]. Similarly to the algebraic approach an
energy function can be calculated in the numerical basis. However, it is not possible to construct a fully orthonormal basis. The number of basis functions is huge. A basis where the occupied orbitals are orthonormal and orthogonal to those in the unoccupied space, and the unoccupied orbitals are non-orthonormal, can be constructed. A transformation that rotates the virtual space into that of the occupied orbitals with out changing the orthonormal properties can also be made (22). By using this generalized Symmetric matrices in the unoccupied space, and the unoccupied orbitals are non-orthonormal. can be avoided.

The optimization equations can be solved using the Newton-Raphson or quasi-Newton methods. The configuration interaction parameters can be optimized in the algebraic approach using the direct CI technique (20,21). The numerical methods are discussed in more detail in ref. (22-24).

The FE-MOFE atomic structure package was used recently in the calculation of the electron affinity of boron (24). In the largest calculation (5 inactive and 4 electrons in the 2s,2p,3s,3p valence shells), the number of configuration state functions (CSF) in Doh symmetry was 105447. The electron affinity obtained was 0.2648(30) eV as compared to the experimental value of 0.277(10) eV.

6. COMPARISON

The FE and algebraic methods are variational, while the FD method is not. In the FD method, the variational feature is used for deriving the Fock equations which then are discretized, while in the FE and algebraic methods the energy functional is first discretized (expanding basis functions) and then varied. By using this generalized matrix and vector equations can be calculated. The optimization equations can be solved using the Newton-Raphson or quasi-Newton methods. The configuration interaction parameters can be optimized in the algebraic approach using the direct CI technique (20,21). The numerical methods are discussed in more detail in ref. (22-24).

In the numerical FD and FE methods, only integrals of the occupied space are needed. All integrals are recalculated after each change of orbitals. By doing this the storage and the computation of the two-electron integrals of the unoccupied space are needed. In the FE and algebraic methods, while it is not obvious how to implement them into the FD method.

The basis of the FD and FE methods cannot be fully orthonormalized because the number of basis functions is huge, while the equations in the algebraic approach are usually solved in an orthonormal basis. The main advantage of the FD and FE methods is the systematic convergence towards the limit of the model with an increasing number of basis functions.

In the FE and algebraic methods the error of the energy is quadratic in the error of the wave function, while in the FD method these errors are of the same order. Therefore it is incorrect to claim that the accuracy of the algebraic approach is comparable to or better than the accuracy of the FD method (26,27) even though the energy may have the same accuracy. The accuracy of a given property, the operator of which does not commute with the Hamilton operator, has the same accuracy as the energy in the FD approximation but not in the algebraic approach. We conclude that the algebraic approach cannot compete with the FE method as far as accuracy is concerned. The comparison is summarized in table 1.

Table 1. A comparison between FD, algebraic and FE methods

<table>
<thead>
<tr>
<th>Property</th>
<th>FD</th>
<th>Algebraic</th>
<th>FE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variational</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Symmetry matrices</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Space matrices</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Automatic optimization</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Second-order convergence</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Number of two-electron Integrals</td>
<td>Yes</td>
<td>N^2</td>
<td>N^2</td>
</tr>
<tr>
<td>Integral transformation</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Well defined convergence</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Increasing size of the basis</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Orthonormal basis</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

*e) n is the number of occupied orbitals and N is the number of basis functions.

References:

Acknowledgements

The research reported in this article has been supported by The Academy of Finland, the Swedish Natural Science Research Council, and the Nordic Council of Ministers. This work has also been supported by IBH Sweden under a joint study contract.
AN ALGORITHM FOR THE LOCATION OF FIRST-ORDER SADDLE-POINT

Culot, G. Dive
Centre d'Ingénierie des Protéines
Université de Liège
Institut de Chimie, B6
B-4000 Liège (Sart-Tilman)
Belgium

VH Nguyen
Département de Mathématiques
Facultés Universitaires de Namur
Rempart de la Vierge, 8
B-5000 Namur
Belgium

I. INTRODUCTION

According to Eyring and Polanyi [1], a chemical reaction coordinate can be seen as a path going from an energy minimum reactant state via a transition state to an energy minimum product state. The transition state is characterized by a maximum energy along the reaction path. This stationary point is maximum along the reaction coordinate and minimum in all other orthogonal directions. Thus, this first-order saddle-point on the potential energy surface.[2] is associated to an indefinite Hessian matrix with only one negative eigenvalue.

The purpose of this paper is to propose an augmented quasi-Newton algorithm to locate a first-order saddle-point. The algorithm is compared to the efficient method of Baker [3] in the study of the methanolysis of protonated methyl-formic-ester.

II. ALGORITHM

A. Quasi-Newton step

The energy \( E(x) \), a function of \( n \) real variables, is, at least, twice continuously differentiable. A quadratic approximation of the energy function around the current point \( x \) can be considered as,

\[
E(x+D) \approx Q(D) = E + G^t D + \frac{1}{2} D^t H D
\]

where \( E, G \) and \( H \) are the energy function, the gradient vector and the Hessian matrix evaluated at the current point \( x \), respectively. \( D \) is a displacement vector around the current point.

The stationary point of \( Q(D) \) is the quasi-Newton step:

\[
D = - H^{-1} G.
\]

This step can be written as,

\[
D = - \sum_{i=1}^{n} \frac{G_i}{b_i} V_i
\]

where \( b_i \) and \( V_i \) are the eigenvalues and eigenvectors of the Hessian \( H \). \( G_i \) is the component of the gradient vector along the eigenvector \( V_i \).

If the Hessian matrix \( H \) is indefinite with one negative eigenvalue, then the quasi-Newton step is a good search direction. But, if the Hessian does not have this expected inertia, then the Hessian matrix has to be perturbed in order to obtain a new step calibration which is ascendent in one direction and descendent along all the orthogonal ones. The new proposed algorithm gives rise to an augmented quasi-Newton step.

B. Augmented quasi-Newton step

The quadratic approximation \( Q(D) \) is only significant near the current point. A scaling of the direction step is then done via a restricted step method. The displacement vector is chosen inside a trust region,

\[
\Omega = \{ D : D^t D \leq R^2 \}
\]

where \( R \) is a trust radius. Inside this trust region, the transition structure search step is calculated via a maximisation of the quadratic approximation along an eigenvector \( V_1 \) and a minimisation of the approximation along the other eigenvectors.

The solution of this optimization problem generates the augmented quasi-Newton step

\[
D(\lambda) = - \frac{G_1}{b_1 - \lambda} V_1 - \sum_{i=2}^{n} \frac{G_i}{b_i + \lambda} V_i
\]

The positive parameter \( \lambda \) is chosen such that
(i) the search direction is inside the trust region \( \Omega \),
(ii) the augmented Hessian matrix is indefinite with one negative eigenvalue. Thus, the conditions

\[
b_1 - \lambda < 0 \quad \text{and} \quad b_i + \lambda > 0, \quad i = 2, \ldots, n
\]
have to be fulfilled. If the Hessian matrix has the expected inertia and if the quasi-Newton step is inside the trust region, then the quasi-Newton step is selected. Otherwise, the step is chosen on the boundary of the trust region with a parameter \( \lambda \) solution of

\[
D'(\lambda)D(\lambda) = R^2 \\
\lambda \geq 0 \\
\lambda > \max\{b_1, -b_2\}
\]

III. APPLICATION

The algorithm is used for the location of a transition state arrangement of the methanolysis of protonated methyl-formic-ester (figure 1). The results are compared to those derived from the method of Baker [3].

\[
\begin{align*}
D'(\lambda)D(\lambda) & = R^2 \\
\lambda & \geq 0 \\
\lambda & > \max\{b_1, -b_2\}
\end{align*}
\]

Figure 1. Methanolysis of protonated methyl-formic-ester. First-order saddle-point located by the augmented quasi-Newton algorithm.

The energy surface associated to this protonated system has a low curvature. From a starting point, which is not taken within the quadratic region of the solution, the algorithm converges to the saddle point (figure 1) after 20 iterations. The eigenvector components associated to the -0.00139 eigenvalue well explain the flip-flop of the water molecule from the hydrated methanol to the methoxy ester group. After 116 iterations, the algorithm of Baker [3] converges to a saddle point in which only the torsional angles of the methyl rotation are concerned. Therefore, this critical point looks like a complex between ester and the couple methanol-water (figure 2).

\[
\begin{align*}
D'(\lambda)D(\lambda) & = R^2 \\
\lambda & \geq 0 \\
\lambda & > \max\{b_1, -b_2\}
\end{align*}
\]

IV. CONCLUSION

This paper deals with an algorithm involved in a transition state arrangement location on an energy surface. This algorithm solves the problem of step estimation as an augmented quasi-Newton displacement by adding a positive shift parameter. The efficiency is well illustrated by a 18-atom system associated with a very low curvature surface.

One important problem inherent to the matrix inertia requires further more investigations. Based on the Hessian inertia, the D step calculation involves a second derivative update which, at present time, has to be improved.

REFERENCE

MOMENTUM SPACE QUANTUM CHEMISTRY

CALCULATIONS. PROBLEMS AND PROMISES.

MIREILLE DEFRANCESCHI

DSM-DRECAM-SPAS, CEN-Saclay,
F-91191 Gif-sur-Yvette Cedex (France)

and

JOSEPH DELHALLE

Laboratoire de Chimie Théorique Appliquée, Facultés
Universitaires N.D. de la Paix, 61, rue de Bruxelles
B-5000 Namur (Belgium).

The model of a molecule in which the nuclei and electrons are assumed to be non-relativistic point charges interacting through electrostatic (Coulomb) forces has been found to provide a satisfactory description of molecular properties [1]. *Ab initio* calculations in molecular quantum chemistry most often mean solving the time-independent Schrödinger equation $H\Psi_r = \epsilon \Psi_r$ where $H$ is the Hamiltonian of a molecule based on that model, $\Psi_r$ is the $i$-th wavefunction and $\epsilon_i$ the corresponding energy eigenvalue. Even when complexity is reduced by considering only the motion of the electrons in a fixed nuclear framework (Born-Oppenheimer approximation), the inherent mathematical difficulties due to the multicenter nature of the electrostatic interactions are such that solutions are not obtainable in explicit form.

Finding suitable and manageable approximate solutions to the electronic Schrödinger equation has thus been a major preoccupation of quantum chemists. Central to attempts at solving such problems is the Hartree-Fock (HF) theory [1]. The essence of this approximation is to replace a complicated many-electron case by a one-electron problem in which the electron-electron interaction is treated in an average way.

Position and momentum HF equations.

Restricted to closed shell systems with $n$ electrons, the $n/2$ doubly occupied HF orbitals $\phi_i$ are obtained in position space as solutions of an integro-differential equation, $(F - \epsilon_i) \phi_i = 0$, where the HF operator $F$ is a one-electron Hamiltonian. It includes a kinetic term and an effective potential itself comprising the electron-nucleus attraction and a Coulombic potential approximating the real electronic interactions. In atomic units, the equation writes as:

$$
\triangledown^2 \phi_i + \phi_i \sum_{i\neq j} Z_j \phi_j / |r_i - r_j| = \epsilon_i \phi_i
$$

(1)

Explicit solutions to eq(1) cannot be obtained because of the terms $|r_i - r_j|^{-1}$ and $|r_i - r_j|^{-2}$. In position space, numerical solutions can be constructed for diatomic molecules, but not for polyatomic systems. In such cases approximate solutions are expressed as truncated linear combinations of basis functions (LCAO expansion). In spite of its successes, the LCAO approximation experiences various difficulties (truncation limits, nature of the basis functions, etc.) which are not entirely controllable [2]. Formulated in momentum space, the HF equations give way to numerical approaches in which Coulombic interactions become tractable even for large atomic molecules [3]. In momentum space, eq(1) becomes:

$$
\frac{p^2}{2}\phi_i + \epsilon_i \phi_i = \frac{1}{2\pi^2} \int dq \left\{ S(q) \sum_{\lambda} \phi^\lambda_\lambda(q) \right\} \phi(p - q)
$$

$$
- \sum_j W_{ij}(q) \phi_j(p - q) = 0,
$$

(2)

where the molecular structure factor $S(q)$ and the interaction terms $W_{ij}(q)$ are:

$$
S(q) = \sum_l Z_l \phi(q R_l)
$$

(3a)

$$
W_{ij}(q) = \int dp \phi_i(p) \phi_j(p - q) e^{i2\pi q \cdot (p R_j - p R_i)}
$$

(3b)

Among other advantages, these equations do not require coordinate systems adapted to the geometry of the molecules to remove Coulombic singularities which make the position space formulation numerically tractable beyond diatomic systems. In eq(2) the only singular contribution comes from the $q^2$ factor.

Numerical Procedure and Problems.

In both position and momentum spaces, iterative procedures are necessary to solve the HF equations. Starting from a trial orbital $\phi_i^{(0)}(p)$, an approximate orbital $\phi_i^{(k+1)}(p)$ is constructed after $k+1$ iterations of eq(2) rewritten as [4]:

$$
\phi_i^{(k+1)}(q) = \frac{\phi_i^{(k)}(q)^{-1}}{2\pi^2} \frac{1}{2} \int dq \left\{ \left[ S(q) - \frac{1}{2\pi^2} \sum_j \phi^\lambda_\lambda(q) \right] \phi_i^{(k)}(p - q) + \sum_j W_{ij}(q) \phi_j^{(k)}(p - q) \right\}
$$

(4)

Numerical and computational problems associated with the implementation of the approach for routine use fall in two main categories: (a) numerical integration, and (b) control of the orthogonality of the numerical orbitals during the iteration steps.

Different integration schemes have been considered to advantageously cancel the singular $q^2$ factor in eq(4) by the integration volume element, Navaza and Tsoucaris [3] have proposed the use of spherical polar coordinates. However, because of the convolution integrals, interpolation schemes are needed in these coordinates since arguments $(p \cdot q)$ do not necessarily belong to the grid points. Another point of view has been to focus on these
convolution integrals and treat them via a more economical fast Fourier transform procedure, but at the expense of an approximate treatment of the $q^2$ singular factor [5,6]. Variants [7,8] based on the Fock transformation [10] have also been proposed to deal with the infinite limits of integration. Computational tests [10] in the case of the helium atom have shown the importance of accuracy and convergence of the integrals and, at present, none of the approaches so far attempted has been satisfactory enough to bring the momentum quantum chemistry calculations beyond a stage of prematurity.

Orthonormalization also raises problems. At each step, the new iterates $\psi^{(k+1)}(p)$ need to be renormalized and orthogonalized to form true canonical HF orbitals [1]. Great care must be exercised in selecting orthogonalization procedures, for instance the so-called Löwdin's symmetric orthogonalization procedure [11], pervasively used in quantum chemistry, mixes all the orbitals simultaneously, tends to contaminate the iterates, and impairs the convergence of the iterative steps [12]. Schmidt orthogonalization does better but looses track of the symmetry of these orbitals. Reformulation of eq(4) in a form with symmetry and orthogonality constraints would be very valuable.

Promises of the Approach.

In spite of the above problems, increasing number of results have been harvested with momentum space quantum chemistry calculations.

Fully numerical HF orbitals for a triatomic molecule have been obtained for the first time [13] with a procedure similar to that originally proposed by Navaza and Tsoucaris [3]. The qualitative and quantitative advantages of using high quality numerical HF orbitals to go beyond the HF level have also been pointed out [14].

With trial orbitals $\psi^{(0)}(p)$ expressed as linear combinations of gaussian functions, it is possible to work out the first iteration and write the first iterates $\psi^{(1)}(p)$ in terms of transcendental functions (e.g. Dawson function); the only numerical steps left being the normalization and orthogonalization. An analysis [15] carried out on the first iterates reveals that, by the virtue of its integral form, eq(4) imparts the right asymptotic behaviour to the first iterates $\psi^{(0)}(p)$ at large and small values of $|p|$. Recent investigations [12,16-18] on atomic systems of increasing complexity (H, He, H', Be, B+, etc.) indeed show significant improvements, qualitative and quantitative, in energy and wavefunction properties. Thus, already with a first step, it is possible to correct for the deficiencies of orbitals expressed as truncated linear combination of basis functions and produced by standard quantum chemistry packages.

Acknowledgments. The authors thank ATOCHEM, CISI, the CEA and the Belgian National Fund for Scientific Research (FNRS) for financial support.

References.

DETERMINATION FROM EXPERIMENTAL MEASUREMENTS OF TRANSPORT COEFFICIENTS AT THE DIFFUSION TIME SCALE IN A TOKAMAK

I. BLUM, H. CAPES, Y. STEPHAN

1 Université de Grenoble, BP53X, 38041 Grenoble Cedex, France.
2 Commissariat à l’Energie Atomique, C.E.N. Cadarache, 13108 Saint Paul lez Durance.
3 CISI INGENIERIE, C.E.N. Cadarache, 13108 Saint Paul lez Durance, France.

Abstract: the knowledge and understanding of the particle and energy transport in a tokamak are of crucial interest to obtain controlled fusion. In fact, the confinement and stability of the plasma cannot be ensured without a whole comprehension of these phenomena. Our aim is then to determine numerically the transport coefficients which govern the equations of conservation from extraneous informations contained in experimental data. This is achieved by a 1D1/2 representation of the resistive MHD equations added to a least square formulation of the constraints. This optimal control problem is then solved using the linear quadratic sequential method and finite elements for space discretization.

1. THE EQUILIBRIUM AND TRANSPORT MODEL

At the diffusion time scale, the momentum conservation equation reduces and the equilibrium assumption holds at each time (cf. Ref [1] for a complete bibliography). In an axisymmetric configuration, it leads to the following 2D Grad-Shafranov equation:

\[ L\Psi = J_t(\Psi) \]  
(1)

where:
- \( \Psi \) is the poloidal flux of the magnetic field \( B \),
- \( L = \frac{\partial}{\partial r} \left( \frac{1}{\mu_0} \frac{\partial}{\partial r} \right) + \frac{\partial}{\partial z} \left( \frac{1}{\mu_0} \frac{\partial}{\partial z} \right) \) is an elliptic operator,
- \( J_t \) is the toroidal plasma current density, and
- \( \mu_0 \) is the magnetic permeability in air.

\[ \text{Fig. 1: Flux lines obtained from IDENTD for an equilibrium at JET tokamak.} \]

The determination of \( J_t \) from experimental data and the numerical resolution of equation (1) with appropriate boundary condition is operated by the software IDENTD (cf. Ref [1][2][3]). This gives the location of the nested magnetic surfaces where \( \Psi \) is a constant as shown in Fig.1.

The averaging of the conservation equations on each magnetic surface leads to the following 1D system (cf. Ref [1]):

Conservation of electrons:

\[ \frac{d}{dt} (V\Psi_e) + \frac{\partial}{\partial \rho} (V^2 \Psi_e) = V<S1> \]  
(2)

where:
- \( \rho = \frac{\phi}{\pi B_0} \) labels the magnetic surfaces,
- \( \phi \) is the toroidal flux of the magnetic field \( B \),
- \( B_0 \) is the magnetic field at a fixed point \( r = R_0 \),
- \( n_e \) is the electronic density,
- \( V = \frac{\partial V}{\partial \rho} \) where \( V \) is the volume enclosed by the magnetic surface \( \rho \),
- and \( <S1> \) is a source term.

Conservation of energy for electrons:

\[ \frac{3}{2} \frac{1}{2} \frac{\partial}{\partial \rho} (\Psi^5/3 \Psi_e) + \frac{\partial}{\partial \rho} (V^2 (\Psi_e + 5kT_e \Psi_e)) = V^3 (\Gamma_e \Psi_e - \alpha (P_e - P_i) + \text{Sohm} + <S2>) \]  
(3)

where:
- \( P_e = n_e kT_e \) is the electronic pressure,
- \( T_e \) is the electronic temperature,
- \( k \) is the Boltzmann constant,
- \( \alpha (P_e - P_i) \) is the equipartition term,
- \( P_i \) is the ionic pressure,
- \( \text{Sohm} \) is a source term due to the Joule effect,
- \( <S2> \) is a source term.

Conservation of energy for ions:

\[ \frac{3}{2} \frac{1}{2} \frac{\partial}{\partial \rho} (\Psi^5/3 \Psi_i) + \frac{\partial}{\partial \rho} (V^2 (\Psi_i + 5kT_i \Psi_i)) = V^3 (\Gamma_i \Psi_i + \alpha (P_e - P_i) + <S3>) \]  
(4)

where:
- \( P_i = n_i kT_i \) is the ionic pressure,
- \( T_i \) is the ionic temperature,
- \( \eta_i \) is the ionic density,
- \( \alpha \) is the resistivity, \( C_2 \) and \( C_3 \) are geometric coefficients,
- \( Z \) is the mean charge of ions;
- \( Q_i = -K_i <V^2 \rho > \frac{\partial T_i}{\partial \rho} \) is the “diagonal” heat flux for ions and \( <S3> \) is a source term.

To determine the resistivity \( \eta_i \), we use the resistive diffusion equation for the flux derivative,

\[ \frac{\partial^2 \Psi'}{\partial t \partial \rho} + \frac{n_i}{\mu_0 \Psi' C_2 C_3} \frac{\partial}{\partial \rho} (\mu_0 C_2 C_3 \Psi') = \frac{\partial}{\partial \rho} \left( \frac{\partial}{\partial \rho} (C_2 C_3 \Psi') \right) = 0 \]  
(5)

and the averaged Grad-Shafranov equation:

\[ \frac{\partial}{\partial \rho} (C_2 C_3 \Psi') = \mu_0 V^2 <\frac{\partial \Psi'}{\partial \rho}>_L \]  
(6)

867
Boundary conditions:

(7)

\[ \frac{\partial n_e}{\partial r}(0,t) = \frac{\partial P_e}{\partial r}(0,t) = \frac{\partial \Psi}{\partial r}(0,t) = 0 \]

(8)

\[ n_e(\rho_{\text{max}},t), P_e(\rho_{\text{max}},t), \Psi(\rho_{\text{max}},t) \]

The validity of our tool has been checked by the following procedure: for given \( D, K_e, K_i, \eta \) and initial state, we first integrate the transport equations and then try to reconstruct the coefficients from initial and final conditions. The behavior of the algorithm has been studied for both exact and perturbed measurements. First results are presently obtained during discharges of the European tokamak TORE SUPRA seated in Cadarache (France).

5. CONCLUSION

Those preliminary results show that the method allows to identify transport coefficients in a way consistent with full 2D equilibrium and experimental data. Great improvements in understanding transport phenomena are made possible and it remains to rely these coefficients to a global physical theory.

Acknowledgement: the authors are grateful to J. Le Foll who is at the origin of this work for many helpful discussions about this problem.

REFERENCES

Abstract - We investigate numerically the nonlinearly saturated single-helicity tearing instability of a visco-resistive current-carrying plane plasma slab using a 2D spectral code. The ratio of viscosity to resistivity is fixed \( v/\eta = 0.2 \). The equilibrium state depends on the \( x \)-coordinate only, perturbations are supposed to be periodic in the \( y \)-direction (period \( L \)) and independent of the \( z \)-direction. The Lundquist number \( S \) is chosen as bifurcation parameter, and we investigate solutions with different fixed values of the period \( L \) by varying \( S \).

Choosing a sufficiently low value of \( L (L = L_0) \), the first branch of the set of solutions bifurcating from the given static equilibrium is numerically found to be stable up to high values of the bifurcation parameter \( S = 10^6 \). Passing to a new value \( L = 2L_0 \), that same branch presents a symmetry breaking. For a period \( L = 4L_0 \), that branch becomes unstable but a lower branch is found stable. Moreover, depending on the choice of the initial conditions, the evolution code may yield spatially very complicated transient states.

I. THE 2D TEARING INSTABILITY

We briefly recall here the physical and mathematical model \([1],[2]\). Let \( \Omega = \mathbb{R}/2, +1/2 \{ x \times y \}, L \) be an open set in \( \mathbb{R}^2 \). The unknowns \( \psi, \phi, \omega \) obey the following equations:

\[
\begin{align*}
\frac{\partial \psi}{\partial t} - \Delta \psi &= S \left( \nabla \cdot \nabla \psi + \nabla_x \frac{\partial \psi_{eq}}{\partial x} \right) \\
\Delta \phi &= \omega \\
\frac{\partial \omega}{\partial t} - \frac{\partial \phi}{\partial \eta} &= \nabla \cdot \left( \frac{\partial \psi_{eq}}{\partial y} \right) &= \nabla \cdot \left( \frac{\partial \psi_{eq}}{\partial x} \right) - \frac{\partial^2 \psi_{eq}}{\partial x^2} \frac{\partial \psi_{eq}}{\partial x} = 0
\end{align*}
\]

where \( \chi = \Delta \psi \) and \( \psi_{eq} \) is a given static equilibrium magnetic flux such that

\[ \psi_{eq}(x) = -(1/\alpha) \log(\cosh(\alpha x)) \]

\( v \) and \( \eta \) are the (constant) viscosity and resistivity, respectively \( (v/\eta = 0.2) \), and \( S \) is the Lundquist number (bifurcation parameter).

The velocity and the magnetic field are related to \( \phi \) and \( \psi \) as follows:

\[ V = \nabla \phi \times \vec{e}_z, \quad B = \nabla \psi \times \vec{e}_z \]

We choose periodic boundary conditions in the \( y \)-direction (sometimes imposing an additional symmetry) and take \( \phi = \Delta \psi = \psi = 0 \) at \( x = \pm 1 \).

For the numerical calculations, \( \chi \) is also an unknown of the problem, with \( \chi = 0 \) at \( x = \pm 1 \).

As already proved \([2]\), we can mathematically justify the existence of bifurcation, making use of compact operators.

The equilibrium magnetic field corresponding to the flux function \( \psi_{eq} \) given above, is parallel to the \( y \)-axis and changes sign at \( x = 0 \).

When the instability sets in, a magnetic island appears (see the representations of curves \( \psi(x,y) = \) constant in the figure).

For the problem of linear stability of the equilibrium we look for solutions expanded in Fourier series.

\[ \exp(iot + i\kappa y) \quad \text{with} \quad m \in \mathbb{N}, \quad k = 2\pi/L \quad \text{or} \quad \pi/L \text{ if we impose symmetries in the} \ y\text{-direction}. \]

A pseudo-spectral code is used to solve our equations, with Fourier-Galerkin decomposition in \( \psi \) and Chebyshev decomposition in \( \chi \), together with implicit or semi-implicit discretization schemes in time. We also impose symmetries in the two directions \( x \) and \( y \).

II. BRANCHES OF STATIONARY SOLUTIONS

II. 1. Structure of the stationary solutions

We consider three different values of the parameter \( k \), namely:

\[ k = 2.5, \quad k = 1.25, \quad k = 0.625 \]

corresponding respectively to lengths \( L, 2L, 4L \) in the \( y \)-direction.

To each number \( m \) of formula (1) we associate a value \( S(m) \) of the parameter \( S \), which corresponds to a bifurcation point where a branch of nonlinear stationary solutions (labelled as "branch m") bifurcates from the equilibrium.
Our main results are represented in the figure and may be summarized as follows:

For the length $L_0$ ($k = 2.5$), the branch 1 has been found to be numerically stable. Taking into account the imposed symmetries, our computation yields only a part of the magnetic island as shown schematically at the bottom of the figure (the lines inside each box represent the separatrix of the magnetic field configuration).

The branch 2 for length $2L_0$ (obtained by symmetry in the $y$-direction from the branch 1 corresponding to length $L_0$) is again numerically stable, but it exhibits a symmetry breaking.

For length $4L_0$, the branch 3 (obtained by symmetry from the branch 2 corresponding to length $2L_0$) is now found to be numerically unstable, with solutions converging in time towards the (stable) solution of the branch 2 of length $4L_0$.

II. 2. Sensitivity to initial conditions

Choosing some initial conditions in the vicinity of a stationary stable solution, we have sometimes observed periodic oscillations in time, with a very slowly decreasing amplitude. Moreover, if we start away from a given stationary stable solution, the evolution of the system presents numerically certain spatially very complicated transient states. These phenomena will be investigated in later work.

References


HIERARCHIC FINITE ELEMENTS-FOR MINDLIN PLATES

LUCIA DELLA CROCE and TERENZIO SCAPOLLA
Dipartimento di Matematica Dipartimento di Matematica
Università di Pavia Università di Pavia
I-27100 Pavia, Italia I-27100 Pavia, Italia

Abstract

The Reissner-Mindlin model describes the deformation of a plate subject to a transverse loading in terms of the normal displacement of the midplane and the rotation of the fibers normal to the midplane. The model is widely used for plates of small to moderate thickness.

It is well known that the numerical approximation of the Reissner-Mindlin plate with standard low degree finite elements leads to solutions that are very sensitive to the plate thickness. For small thickness the numerical solution is very far from the exact one. The phenomenon is referred to as locking. Several non-standard finite element spaces and techniques have been devised to overcome such a difficulty. We recall, among others, reduced and selective integration, interpolation and projection techniques combined with non-classical formulations like mixed or hybrid approaches. As far as we know only low order element, generally with degree 1 or 2, have been tested for Reissner-Mindlin plates, both for standard and non-standard formulations.

In recent years high order finite elements, known as $p$ finite elements, have been introduced and successfully applied in several fields, e.g., for elasticity and Kirchhoff plate problems. High order elements have shown to be robust and able to absorb locking phenomena in the case of nearly incompressible materials. In this note we present some numerical results obtained with a family of hierarchic finite elements with degree from 1 to 4 for the solution of Reissner-Mindlin plates in the standard formulation. We show that the locking of the solution does not appear when high order elements are used.

1. Derivation of the Reissner-Mindlin model

Let us consider a three-dimensional body occupying a region $V$. We assume that the body under consideration is isotropic.

The potential (or strain) energy per unit volume is given by

$$W = \frac{1}{2} \lambda \left( e_{zz} + e_{yy} + e_{xx} \right)^2$$

$$+ \mu \left( e_{zz}^2 + e_{yy}^2 + e_{xx}^2 + 2 e_{yz}^2 + 2 e_{xz}^2 + 2 e_{xy}^2 \right)$$

where $e_{zz}, e_{yy}, e_{xx}, e_{yz}, e_{xz}, e_{xy}$ are the components of strain and $\lambda$ and $\mu$ are the Lamé coefficients, constant characterizing the elastic behaviour of the body. The coefficients can be expressed in terms of the modulus of elasticity $E$ and the Poisson's ratio $\nu$:

$$\lambda = \frac{E\nu}{(1 + \nu)(1 - 2\nu)}$$

$$\mu = \frac{E}{2(1 + \nu)}$$

The components of stress at a given point are linear and homogeneous functions of the components of strain at the same point and vice versa (see, e.g., [9]). Denoting by $u, v, w$ the components of displacement we have the relations between strain and displacement:

$$e_{xx} = \frac{\partial u}{\partial x}$$

$$e_{yy} = \frac{\partial v}{\partial y}$$

$$e_{zz} = \frac{\partial w}{\partial z}$$

$$e_{yz} = \frac{1}{2} \left( \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \right)$$

$$e_{xz} = \frac{1}{2} \left( \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right)$$

$$e_{xy} = \frac{1}{2} \left( \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right)$$

In the Reissner-Mindlin theory two new fields $\phi_1$ and $\phi_2$ are introduced. These fields represent the rotation of the cross-sectional planes to which, respectively, the $z$-axis and $y$-axis is normal:

$$u = -z \phi_1(x, y)$$

$$v = -z \phi_2(x, y)$$

$$w = w(x, y)$$

Through the Reissner-Mindlin hypotheses the total strain energy becomes:

$$U(w, \phi_1, \phi_2)$$

$$= \frac{1}{2} \frac{E\nu^2}{12(1 - \nu^2)} \int \left( \phi_{1z}^2 + \phi_{2y}^2 + 2\nu \phi_{1z} \phi_{2y} \right) + \frac{1 - \nu}{2} \left( \phi_{1z}^2 + \phi_{2y}^2 \right) dz dy$$

$$+ \frac{1}{2} \frac{E\nu k}{2(1 + \nu)} \int \left( (w_{xy} - \phi_1)^2 + (w_{y} - \phi_2)^2 \right) dz dy$$

where $k$ is the shear correction factor.

Taking into account the external load $p$ we get the following variational formulation for the Reissner-Mindlin model for plates:
U\left(\psi_1, \psi_2; v, \psi_1, \psi_2\right) = \frac{E\psi_0}{12(1-\nu^2)} \int_\Omega \left( \psi_1 v_1 + \psi_2 v_2 + \nu \left( \psi_1 v_1 + \psi_2 v_2 \right) \right) \, dx \, dy + \frac{E t k}{2(1+\nu)} \int_\Omega \left( \psi_1 - w_{12} \right) \left( \psi_1 - w_{12} \right) \, dx \, dy + \int_\Omega \psi_0 \, dx \, dy.

Setting \hat{\delta} = (\phi_1, \phi_2)$, the previous problem can be stated in the following way:

\[
\begin{align*}
\text{Find } (\hat{\delta}, w) \in (H^2_0)^2 \times H^1_0 \text{ such that } \\
\frac{E\psi_0}{12(1-\nu^2)} A(\delta, \psi) + \frac{E t k}{2(1+\nu)} \|\nabla w - \hat{\delta}\|_0^2 = (p, v) \\
\forall (\delta, v) \in (H^2_0)^2 \times H^1_0
\end{align*}
\]

where

\[
A(\delta, \psi) = \int_\Omega \left( \psi_1 v_1 + \psi_2 v_2 + \nu \left( \psi_1 v_1 + \psi_2 v_2 \right) \right) \, dx \, dy + \frac{1-\nu}{2} \int_\Omega \left( \psi_1 + \phi_2/2a \right) \left( \psi_1 + \psi_2/2a \right) \, dx \, dy.
\]

2. Numerical approximation

One of the advantages of the Reissner-Mindlin approach is the fact that the variational formulation allows the use of continuous ($C^0$) finite elements, since only first derivatives appear. We recall that in the Kirchhoff model conforming approximations require $C^1$ finite elements. It is well known that the numerical approximation of the Reissner-Mindlin plate with standard low degree finite elements leads to solutions that are very sensitive to the plate thickness. For small thickness the numerical solution is far from the exact one. The phenomenon is referred to as locking. Several non-standard finite element spaces and techniques have been devised to overcome such a difficulty. We recall, among others, reduced and selective integration, interpolation and projection techniques combined with non-classical formulations like mixed or hybrid approaches. As far as we know only low order elements, generally with degree 1 or 2, have been tested for Reissner-Mindlin plates, both for standard and non-standard formulations.

In recent years high order finite elements, known as $p$ finite elements, have been introduced and successfully applied in several fields, e.g., for elasticity and Kirchhoff plate problems (see [1,2,3]). High order elements have shown to be robust and able to absorb locking phenomena in the case of nearly incompressible materials (see [4]). In this way we present some numerical results obtained with a family of hierarchic finite elements for the Reissner-Mindlin plate in the standard formulation. Hierarchy of finite elements means that shape functions are added to increase the degree of approximation, leaving unchanged the previous functions. The functions used are based on the family of Legendre polynomials. This class presents good properties from the point of view of roundoff error accumulation with respect to the increase of the polynomial degree. The family consists of finite elements of degree from 1 to 4.

Referring to the classification of the shape functions suggested by Babuška [1] as nodal, side and internal functions, we give in Table 1 the number of shape functions of each type for a general value of the degree $p$.

<table>
<thead>
<tr>
<th>$d$</th>
<th>nodal f.</th>
<th>side f.</th>
<th>internal f.</th>
<th>total $#$ f.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>4</td>
<td>-</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>8</td>
<td>-</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>12</td>
<td>1</td>
<td>17</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>16</td>
<td>3</td>
<td>23</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>20</td>
<td>6</td>
<td>30</td>
</tr>
<tr>
<td>$p$</td>
<td>$4(p-1)$</td>
<td>$\frac{1}{2}(p-2)\times$</td>
<td>$\frac{1}{2}(p-2)(p-3)$</td>
<td>$4p$</td>
</tr>
</tbody>
</table>

Table 1: Number of shape functions for different degrees

We have written a code where shape functions with degree from 1 to 4 are used. The hierarchic structure of the functions allows easy extensions to higher degrees.

We introduce two finite dimensional spaces $\Phi$ and $V$, respectively for the discrete rotations and displacement. The approximate problem can be stated in the following way:

\[
\begin{align*}
\text{Find } (\hat{\delta}^h, w^h) \in \Phi \times V \text{ such that } \\
\frac{E\psi_0}{12(1-\nu^2)} A(\delta^h, \psi^h) + \frac{E t k}{2(1+\nu)} \|\nabla w^h - \hat{\delta}^h\|_0^2 = (p, v^h) \\
\forall (\delta^h, v^h) \in \Phi \times V
\end{align*}
\]

where

\[
A(\delta^h, \psi^h) = \int_\Omega \left( \psi_1 v_1 + \psi_2 v_2 + \nu \left( \psi_1 v_1 + \psi_2 v_2 \right) \right) \, dx \, dy + \frac{1-\nu}{2} \int_\Omega \left( \psi_1 + \phi_2/2a \right) \left( \psi_1 + \psi_2/2a \right) \, dx \, dy.
\]
3. Numerical results

A series of numerical experiments on several plates with different physical data has been performed. Due to the lack of space only a few of them are presented. Comparison are made with the theoretical results computed by Timoshenko [6].

For each test, among others, displacement at the center $C$ of the plate and the strain energy have been computed. Let $w(C)$ denote the exact displacement at the center of the plate and $w_h(C)$ the finite element solution. The relative displacement error is defined as

$$d = \frac{w(C) - w_h(C)}{w(C)} \times 100.$$  

The exact strain energy was not available. Out of the discrete strain energy an extrapolation has been made in order to get an accurate value of the energy. Let $E$ denote such a energy, let $E_h$ be the discrete energy. The relative energy norm $||e||$ of the error $e = w - w_h$ can be expressed in the following way:

$$||e|| = \left( \frac{E - E_h}{E} \right)^{1/2} \times 100.$$  

We consider a square plate with uniform decomposition. On the boundary the plate is clamped. Due to the symmetry we have solved the problem on a quarter of plate. We have considered a unit plate with thickness 0.1 (thick plate), 0.01, 0.001, 0.0001. The last value is related to an ultrathin plate. We have considered this value in order to assess the stability that the locking is larger when small thicknesses are taken into account.

In Fig.1a-d we consider a plate with thickness $t = 0.01$. The relative error for the displacement vs. number of degrees of freedom is shown. The results are given for values of the degree $p = 1, 2, 3, 4$.

The previous results are related to a moderately thick plate. The pictures show that for $p = 1$ the rate of convergence is very slow and the solution is very inaccurate. The numerical solution exhibits a strong locking. For $p = 2, 3, 4$ we note a good behaviour. The respective displacement solutions approach the exact solution with a relatively small number of degrees of freedom. As expected, the better convergence is achieved for increasing values of $p$. 

In Fig. 2a-d we consider a plate with thickness 0.001. We show, in log-log scale, the relative energy norm error vs. number of degrees of freedom. The results are given for values of the degree $p = 1, 2, 3, 4$.

In the previous pictures a thin plate is considered. The discrete strain energy is a good global indicator since it takes into account a distributed solution. For $p = 1$ we observe no convergence. The behaviour is far worst than the case with thickness $t = 0.01$.

In Fig. 3a-d we consider, for a fixed degree of approximation $p = 2$, different values of the thickness. We show, in log-log scale, the relative error for the displacement vs. number of degrees of freedom.
In Fig. 4a–d we consider, for a fixed degree of approximation $p = 4$, different values of the thickness. We show, in log-log scale, the relative error for the displacement vs. number of degrees of freedom.

In the previous pictures we consider the numerical approximation with fixed degree $p = 2$ and we change the value of thickness. The sequence of figures well shows that the convergence slows down noticeably when the thickness is reduced. The log-log scales allow to evaluate the different rates of convergence (note the different scales on the $y$-axis). In particular, for the smaller thickness, the locking of the numerical solution is well exhibited.

We now consider a fixed degree $p = 4$ and we change the thickness. We observe that, with such an high order element, convergence is achieved even for the smaller value of thickness $t = 0.0001$. This shows that the high order element is able to absorb the phenomenon of locking found for lower degrees (see Fig. 3a–d).


Abstract. A finite difference method is used to solve the system of 2-D elastic wave equations. The solution is expressed as a displacement of a particle from its original position. By means of computer graphics we describe the solution as a 3-D wave surface, and describe its propagation by real time animation. In this way, we can observe the phenomena described by the system of equations more easily and in a much more obvious fashion than with conventional methods. Here, we use an explicit method to solve the equations in time, this makes real time calculation, modeling and rendering possible. The algorithm has been implemented on an Iris-AD personal graphics workstation.

1. Introduction

Computer graphics provides an excellent means for exploring the elastic waves visually. Here we shall give an experimental procedure for modeling and rendering the elastic wave and its propagation in real time. To make things easier for demonstration we choose some typical parameters for the wave equation. That means that the solution for each function has been separated into two independent groups along each axis, the group of even points and the group of odd points. The value at one even point only affects other even points. The value at one odd point only affects other odd points. For the propagation of the wave, this phenomenon means that if an external body force acts at an even point it will only propagate among the even points. Equation (5) and Diagram I can explain why this happens to u and v. The equations for u and v given by (5) in the x direction are considered here. We suppose that the u wave begins at \( u_{lm} \) and the v wave begins at \( v_{lm} \), then they affect\( u_{lm+1} \) and \( v_{lm+1} \) at the next time step. After two time steps the waves only propagate to \( u_{lm+2} \) and \( v_{lm+2} \) and \( v_{lm+1} \), nothing will be affected at \( u_{lm+1} \) and \( v_{lm+1} \) from equation (6). Therefore, here we shall use these kind of solutions to demonstrate our visualization on taking only the even points or only the odd points, depending on where the external body force acts. But observe that because of this uncoupling, the Courant Friedrichs Lewy condition is not satisfied and thus these solution are not truly physical (3). We will discuss this in more detail in a later paper.

2. Finite-difference method

Here we use the second order approximation for the first derivatives.

\[
u^m(l+1) = u(x(t+1)) = \frac{u(x(t)) - u(x(t-1))}{2h}
\]

Let

\[
u^m_{lm} = u(x, y, t, \Delta x, \Delta y, \Delta t), \Delta x = \Delta y = \Delta t = k, \quad r = k/h, \quad 0 < L \leq k, \quad 0 < m < M, \quad r > 0.
\]

Then (3) can be expressed as an explicit system

\[
u^{n+1}_{lm} = \nu^{n}_{lm} + (rtB)f_{l+1}^{n+1} - \nu^{n}_{lm} - \nu^{n}_{lm+1} + 2k f_1^{n+1}(nk)
\]

\[
u^{n+1}_{lm} = \nu^{n}_{lm} + (rtB)f_{l+1}^{n+1} - \nu^{n}_{lm} - \nu^{n}_{lm+1} + 2k f_2^{n+1}(nk)
\]

\[
u^{n+1}_{lm} = \nu^{n}_{lm} + r(e+2w)(\nu^{n}_{lm+1} - \nu^{n}_{lm}) + r(e+2w)(\nu^{n}_{lm+1} - \nu^{n}_{lm})
\]

\[
u^{n+1}_{lm} = \nu^{n}_{lm} + w r \left[ (\nu^{n+1}_{lm+1} - \nu^{n+1}_{lm}) + (e+2w)(\nu^{n+1}_{lm+1} - \nu^{n+1}_{lm}) \right]
\]

(5)

Using the Trapezoidal rule, we get the displacement U and V as

\[
U^{n+1}_{lm} = U^{n}_{lm} + 0.5k \left( u^{n}_{lm+1} + u^{n}_{lm} \right)
\]

\[
V^{n+1}_{lm} = V^{n}_{lm} + 0.5k \left( v^{n}_{lm+1} + v^{n}_{lm} \right)
\]

(6)

Here we should observe that the solution points of all the functions U, V, u, v, g, h are uncoupled in space domain, but using this method they can be divided into two coupled sets of points. This means that the solution for each function has been separated into two independent groups along each axis, the group of even points and the group of odd points. The value at one even point only affects other even points. The value at one odd point only affects other odd points. For the propagation of the wave, this phenomenon means that if an external body force acts at an even point it will only propagate among the even points. Equation (5) and Diagram I can explain why this happens to u and v. The equations for u and v given by (5) in the x direction are considered here. We suppose that the u wave begins at \( u_{lm} \) and the v wave begins at \( v_{lm} \), then they affect \( u_{lm+1} \) and \( v_{lm+1} \) at the next time step. After two time steps the waves only propagate to \( u_{lm+2} \) and \( v_{lm+2} \) and \( v_{lm+1} \), nothing will be affected at \( u_{lm+1} \) and \( v_{lm+1} \) from equation (6). Therefore, here we shall use these kind of solutions to demonstrate our visualization by taking only the even points or only the odd points, depending on where the external body force acts. But observe that because of this uncoupling, the Courant Friedrichs Lewy condition is not satisfied and thus these solution are not truly physical (3). We will discuss this in more detail in a later paper.

\[
u^{n+1}_{lm} + u^{n+1}_{lm} + (rtB)f_{l+1}^{n+1} - \nu^{n}_{lm} - \nu^{n}_{lm+1} + 2k f_1^{n+1}(nk)
\]

\[
u^{n+1}_{lm} + v^{n+1}_{lm} + (rtB)f_{l+1}^{n+1} - \nu^{n}_{lm} - \nu^{n}_{lm+1} + 2k f_2^{n+1}(nk)
\]

\[
u^{n+1}_{lm} + \nu^{n+1}_{lm} + r(e+2w)(\nu^{n}_{lm+1} - \nu^{n}_{lm}) + r(e+2w)(\nu^{n}_{lm+1} - \nu^{n}_{lm})
\]

\[
u^{n+1}_{lm} + \nu^{n+1}_{lm} + w r \left[ (\nu^{n+1}_{lm+1} - \nu^{n+1}_{lm}) + (e+2w)(\nu^{n+1}_{lm+1} - \nu^{n+1}_{lm}) \right]
\]

(5)

Using the Trapezoidal rule, we get the displacement U and V as

\[
U^{n+1}_{lm} = U^{n}_{lm} + 0.5k \left( u^{n}_{lm+1} + u^{n}_{lm} \right)
\]

\[
V^{n+1}_{lm} = V^{n}_{lm} + 0.5k \left( v^{n}_{lm+1} + v^{n}_{lm} \right)
\]

(6)

Diagram 1
initial values are chosen as zero. Two kinds of boundary conditions are chosen here. One is the free boundary condition, or zero Neumann boundary conditions. The other the fixed boundary condition, the usual zero Dirichlet boundary condition.

Initial values:

\[ \text{At} t=0 \quad \text{and} \quad \text{at} \quad \text{and} \quad \text{all} \quad \text{initial} \quad \text{values} \quad \text{are} \quad \text{zero}. \]

Boundary conditions:

A. Free boundary:

\[
\begin{align*}
U_1^0 &= U_{1,1}^0, & \quad U_0^0 &= U_{1,1}^0, & \quad V_1^0 &= V_{1,1}^0, & \quad V_0^0 &= V_{1,1}^0, \\
U_m^0 &= U_{m,1}^0, & \quad U_0^m &= U_{1,1}^m, & \quad V_m^0 &= V_{1,1}^m, & \quad V_0^m &= V_{1,1}^m.
\end{align*}
\]

B. Fixed boundary:

\[
\begin{align*}
U_1^0 &= U_{1,1}^0, & \quad V_1^0 &= V_{1,1}^0, \\
U_0^0 &= U_{1,1}^0, & \quad V_0^0 &= V_{1,1}^0.
\end{align*}
\]

6. Example

The area of interest chosen here is \(0 < x < 1, 0 < y < 1\), so that \(M = 1, L = 1\). Fixed boundary condition (10) and external body force function (11) are used. Pictures 1 - 8 show the vertical displacement function \(V(x, t)\) at the times for \(n = 1, 2, 4, 8, 16, 48, 64, 80, 112, 160, 192, 256\). Here we see that the wave begins from the lower left corner and propagates towards the upper right corner.

7. Conclusion

We have shown a way to visualize elastic wave propagation with real time animation by means of computer graphics. This improves interpretation of the numerical results. We can use this method to show how the different media or different external forces affect on the elastic waves and how elastic waves propagate, as well as how different parameters, boundary conditions or initial conditions affect the solutions.
ABSTRACT - The paper deals with mathematical modelling of industrial continuous distillation column for separation of four-component mixture composed of furfural, methanol, acetic acid and water. The aim of the work is optimisation of existing device in the sense of product quality improvement and minimisation of energy consumption. The model was examined by the aid of digital simulation language SIMCOS. Presented results show that the best solution lies in construction change which have good influence especially to the bottom product and so to the energy consumption of the device.

1. Plant description

Presently the input mixture is fed to the device at the 25th tray and furfural is taken out at the 28th tray as a side stream product. The second product is methanol which is separated in the upper part of the column. As it is shown in Figure 1, furfural mixture is kept in separating reservoir leaving the device with flow rate \( D_A \). Furfural is heterogeneous azeotrope. In mentioned reservoir it separates to heavier component, which leaves the system with flow 0 and represents one of the final products while lighter component is taken to the bottom reservoir where it mixes with the industry environment demanding the vapor flow rate \( G \) to be enlarged for 10 to 15%.

2. Mathematical modelling and simulation of existing device

It is obvious that the most important variables of the system are input and output flow rates and their compositions. So we started modelling procedure by writing component and mass balances for each tray of the column (for the simplicity of notation the condenser is numbered as the 44th plate):

\[
\frac{d}{dt} x_{j1}(t) = L_{j1}(t) x_{j+11}(t) - L_j(t) x_{j1}(t) + V_{j1}(t) y_{j1}(t) - V_j(t) y_{j1}(t) +
\]

\[
+ L_{sij1}(t) x_{sij1}(t) - L_{soj1}(t) x_{soj1}(t) + V_{sij1}(t) y_{sij1}(t) - V_{soj1}(t) y_{soj1}(t)
\]

\( j = 1, \ldots, 44; \quad i = 1, \ldots, 4 \)

where \( L(t) \) and \( V(t) \) denotes the liquid and vapor flow rates from the \( j \)-th tray, \( x_{j1}(t) \) the liquid composition of \( i \)-th component on \( j \)-th tray and \( y_{j1}(t) \) the vapor composition of \( i \)-th component upon \( j \)-th tray, \( L_{sij1}(t) \), \( L_{soj1}(t) \), \( V_{sij1}(t) \) and \( V_{soj1}(t) \) are input and output liquid and vapor flow rates on \( j \)-th tray with compositions \( x_{sij1}(t) \), \( x_{soj1}(t) \), \( y_{sij1}(t) \) and \( y_{soj1}(t) \).

A lot of methods can be used to describe the relationship between liquid and vapor compositions of mixture at defined pressure and temperature. Especially in the absence of reliable vapor-liquid equilibrium data the following equations are very popular in chemical engineering:

\[
y_{j1}(t) = \frac{K_j x_{j1}(t)}{y_{j1}(t)}; \quad j = 1, \ldots, 43;
\]

where \( K_{j1} \) is distribution coefficient of \( i \)-th component at \( j \)-th tray of the column.

The analysis of measured data and simulation results, which we obtained by the aid of digital simulation language SIMCOS, gave the following conclusions:

- Concentration of methanol in the flow rate \( D_M \) is higher than the prescribed value so the \( G \) could be reduced.
- Concentration of furfural in the flow rate \( D_A \) is inside the prescribed values.
- Concentration of furfural in bottom flow \( B \) is too high thus demanding the vapor flow rate \( G \) to be enlarged for 10 to 15%.

( Normal value: furfural concentration in bottom product were approximately 0.17%. Due to more rigorous ecological demands this values have to be reduced to 0.05 - 0.1%. Simulation results and experiments on the plant have shown that for such improvement vapor flow rate \( G \) should be enlarged for 10 to 15%.)
These conclusions indicate that vapor flow rate $G$ should be enlarged because of ecological demands meaning that energy consumption will of course be also greater. Nothing really effective could be therefore made with this column structure.

3. Structural rearrangement of distillation column

In the second step of modelling we decided to examine some possibilities in structural rearrangement of the device. The best result in this phase was given by the solution in Figure 1. The column is now fed back with the lighter furfural component from the separation reservoir on the 26th tray. This of course changed mass balances of 26th tray and reservoir in which $\phi_{in}$ is entering. The consequence is, that also feed flow rate $F$ and its concentration is changed. The steady state values of this changes can be evaluated from previous equations.

The simulation was made for changed working conditions of the plant and results are shown in Figures 2 to 4 for input and output flow rates, furfural and methanol. From this we can see that the steady state composition of all trays in the column are changed and that this rearrangement caused the lowering of concentrations of furfural in the bottom flow $B$ to the prescribed values, leaving the other requirements fulfilled.

4. Conclusion

Experiments on the plant showed that the proposed assumptions were justified. We also assume that the model could benefit with the introduction of nonconstant distribution coefficients, what will be needed in later work for the improvement of the existing control loops. Digital simulation language SIMCOS we used was very efficient tool for solving complex mathematical models with a large number of variables, nonlinearities and measured data and saved a lot of time and work on the real device.

References:


NONLINEAR FINITE ELEMENT ANALYSIS OF SLOW CRACK GROWTH

XIANGQIAO YAN, SHANYI DU and DUO WANG
Harbin Institute of Technology, Harbin, 150006, P.R.China

ABSTRACT—In this paper, a material-nonlinear finite element program of the quasi-three dimension problem developed by authors of this paper is described from a few respects. Further, it is illustrated how this program is used to analyze plane problems. Finally, a finite element analysis of the process of slow crack growth is made for a center-cracked specimen subjected to monotonically increased load until the point of fast fracture is reached. Numerical results and experimental results are compared. Variation laws of some fracture mechanics parameters are given with the stable crack growth.

I. INTRODUCTION

In a cracked body with toughened materials or in the plane-stress state, the crack will grow monotonically and slowly increased load before the fast fracture is reached. Many scholars attracted their attention to it. They have done a lot for it [11-17]. Experiments [8, 9] showed that there is the stable delamination crack growth in some stacking sequence composite laminates. Mahishi [10] ever pointed out that while considerable progress has been made in understanding the delamination mechanisms in composites incorporating brittle or quasi-brittle matrix materials, the problem of toughened polymer matrix or metal matrix composite, in which large scale yielding is associated with the cracking, requires special attention.

In view of these, a material nonlinear finite element program of the quasi-three dimension problem was recently developed by the authors [11]. Using this program, not only the nonlinear stress analyses, the delamination onset and the stable delamination crack growth for composite laminates but the nonlinear finite element analysis of stable crack growth process for plane cracked body are made. Here, the program is described from a few respects and a finite element analysis for a center-cracked specimen subjected to monotonically and slowly increased load until the point of fracture is reached is taken for example to check it. Finally, the variation laws of some fracture mechanics parameters are given with the stable crack growth.

II. DESCRIPTIONS OF PROGRAM

Here, a material nonlinear finite element program [11] of the quasi three dimension problem developed by the authors will be described from a few respects.

A. GEOMETRICAL DESCRIPTION

The origin of the quasi-three dimension problem can refer to the Ref [13]. Its schematic illustration and simplified model are shown in Fig. 1. Its displacement field expressions are listed as follows [13]:

\[
\begin{align*}
u &= \epsilon_0 u(y, z) \\
v &= v(y, z) \\
w &= w(y, z)
\end{align*}
\]

where \(\epsilon_0\) is uniform action strain in the direction of \(x\).

Strain-displacement relations are

\[
\begin{align*}
\epsilon_x &= u_x, \\
\epsilon_y &= v_y, \\
\epsilon_z &= w_z
\end{align*}
\]

\[
\begin{align*}
\gamma_{yz} &= v_z + w_y, \\
\gamma_{xy} &= w_x + u_z, \\
\gamma_{xz} &= u_y + v_x
\end{align*}
\]

It can be seen that the strains corresponding to displacement expressions (2-1) can be classified into two parts, one part not depending on the coordinates, i.e.

\[
[\bar{\epsilon}] = \begin{bmatrix} \epsilon_{0x} & 0 & 0 & 0 & 0 \end{bmatrix}^T \quad (2-3)
\]

and that total strains are

\[
[\epsilon] = [\bar{\epsilon}] + [\bar{\epsilon}]
\]

According to the displacement finite element method, we have

\[
\begin{align*}
U &= \sum N_i U_i \\
V &= \sum N_i V_i \\
W &= \sum N_i W_i
\end{align*}
\]

where \(N_i\) are interpolation functions [14], \(U_i, V_i\) and \(W_i\) nodal displacements.

Introducing the symbols:

\[
[\bar{x}] = [\bar{x}]_i = (i=1, 2, \ldots, 5)
\]

then the following results can be obtained from formulas (2-2) and (2-4):

\[
[\bar{\epsilon}] = [B][\bar{\delta}],
\]

where \([\bar{\delta}]^T\) is the element displacement vector, \([B]\), for a triangle element, is the (5x5) strain matrix whose expression is deleted.

B. MATERIAL-NONLINEAR FINITE ELEMENT EQUATIONS

Here, it is supposed that the individual lamina material in composite laminates is homogeneous, orthotropic and after yielding, follows Hill's orthotropic plasticity theory, referred to the Ref [12].

In a material coordinate system, incremental stress-strain relations can be written as [12]:

\[
[d\sigma] = [C]_p [d\epsilon]
\]

By coordinate translation for (2-6), incremental
stress-strain relations expressed in the structural coordinate system can be obtained as follows:

\[ [d \sigma] = [C]_{np} [d \epsilon] \]  
(2-7)

where

\[ [d \sigma] = [T]^{-1} [d \sigma] \]
\[ [d \epsilon] = [T]^{-1} [d \epsilon] \]
\[ [C]_{np} = [T]^{T} [C]_{np} [T] \]

and \([T]\) is the translation matrix whose expression is deleted here.

Introducing the following symbols:

\[ [d \sigma]^i = [d \sigma]_{11} \quad (i=1,2,\cdots,5) \]  
(2-8)
\[ [d \epsilon]^i = [d \epsilon]_{11} \]
a series of derivations are made to find

\[ [d \sigma]^i = [d \sigma]_{11} \quad (i=1,2,\cdots,5) \]  
(2-9)

in which

\[ (\sigma_{a1})_{np} = (C_{a11}, n, 11) \quad (a, n=1,2,\cdots,5) \]  
(2-10)

and \( d \sigma \) expresses the increment of \( \sigma_{0} \).

It is noted that, in a material coordinate system, the elastic-plastic matrix \([C]_{np}\) can be expressed in terms of the elastic \([C]_{p}\) and the plastic matrix \([C]_{p}\), i.e.

\[ [C]_{np} = [C]_{p} - [C]_{p} \]  
(2-11)

Similarly, in a structural coordinate system, the elastic-plastic matrix \([C]_{np}\) can also be expressed in terms of the elastic matrix \([C]_{p}\) and the plastic matrix \([C]_{p}\), i.e.

\[ [C]_{np} = [C]_{p} - [C]_{p} \]  
(2-12)

Introducing the symbols:

\[ (\bar{\sigma}_{a1})_{np} = (C_{a11}, n, 11) \quad (a, n=1,2,\cdots,5) \]  
(2-13)

then the following results can be obtained from formulas (2-10) and (2-12)

\[ (\bar{\sigma})_{np} = (\bar{\sigma})_{p} - (\bar{\sigma})_{p} \]  
(2-14)

By substituting (2-14) into (2-9), one can find

\[ [d \sigma]^i = [d \sigma]_{11} [\bar{\sigma}]_{0} [d \epsilon]^i \]  
(2-15)

in which

\[ [d \sigma]^i = [d \sigma]_{11} [\bar{\sigma}]_{0} [d \epsilon]^i \]  
(2-16)

and

\[ [d \sigma]^i = [C]_{12} [C]_{13} [C]_{14} [C]_{15} [C]_{16} [d \sigma]_{0} \]  
(2-17)

According to the formulas (2-15), we know, if the \([d \sigma]_{0}^i\) are regarded as original stresses, then the material-nonlinear problem is translated into the material-linear problem possessing the original stresses \([d \sigma]_{0}^i\). It can be seen from formulas (2-16) and (2-17) that the original stresses \([d \sigma]_{0}^i\) depend on not only the stress state before immediate loading but the strain \([d \epsilon]^i\) due to immediate loading. Thus, the original stress vector \([d \sigma]_{0}^i\) translated by original stresses \([d \sigma]_{0}^i\) depends on the immediate strain increment \([d \epsilon]^i\). The following finite element equations can be derived by using stress-strain relations (2-15), strain-displacement relations (2-5) and the variation principle:

\[ [K]_{i1} [d \sigma]_{1} = (d \bar{\sigma})_{1} + \gamma (d \bar{\sigma})_{1} \]  
(2-18)

where the subscript \(i\) expresses the \(i\)th step loading increment.

\[ [K] = \text{the elastic structural stiffness matrix, in which an element stress vector is} \]

\[ (d \bar{\sigma})_{1} = f (B)^{T} (d \sigma)_{0} \]  
(2-19)

\[ (d \bar{\sigma})_{1} = \text{the displacement vector due to the \(i\)th step loading increment;} \]

\[ (d \sigma)_{0} = \text{the load vector due to the \(i\)th step loading increment;} \]

\[ (d \sigma)_{0} = \text{the original stress vector translated by \((d \sigma)_{0}\) in which an element \(\sigma_{0}\) original stress vector is} \]

\[ (d \sigma)_{0} = f (B)^{T} (d \sigma)_{0} \]  
(2-20)

Because the original stress vector \([d \sigma]_{0}\) depends on the immediate strain increment \([d \epsilon]_{0}\), and the immediate strain increment \([d \epsilon]_{0}\) is also unknown, equations (2-18) must be solved by using an iterative method. After finding the approximate \([d \sigma]_{0}\), the calculation formula of \([d \sigma]_{0}\) is

\[ (d \sigma)_{0} = f (B)^{T} (d \sigma)_{0} \]  
(2-21)

C. LOADING AND UNLOADING CRITERIA

Since the local unloading around the crack tip due to stable crack growth occurs, loading and unloading problem are necessarily considered in simulating the stable crack growth.

The yielding equation for a homogenous, orthotropic material can be expressed in terms of the effective stress

\[ 2g(\sigma_{11}, h) = \frac{2}{3} (E_{0} + G_{0} + h_{0}) \frac{\sigma^{2} - h^{2}}{2} = 0 \]  
(2-22)

where what individual symbols in (2-22) mean can refer to the Ref [12].

Introducing the following symbols:

\[ f(u_{11}, h') = \frac{3}{E_{0} + G_{0} + h_{0}} g(u_{11}, h) \]  
(2-23)

in which

\[ h' = \frac{h^{2}}{2} \frac{2}{E_{0} + G_{0} + h_{0}} \]  
(2-24)

then the yielding equation can be expressed as

\[ f(\sigma_{11}, h') = \frac{\sigma^{2} - h'^{2}}{2} = 0 \]  
(2-24)
where the original value of \( h^{'}, h^{``} \), is
\[
\begin{align*}
  h^{'} &= \frac{3}{2(E_{0}+G_{0}+H_{0})} \\
  h^{``} &= \sigma_{y}
\end{align*}
\]  
(2-25)

The following relations for a homogeneous isotropic material exist:
\[
3E_{0}=3G_{0}=3H_{0}=L_{0}=N_{0}=H_{0}
\]  
(2-26)

and
\[
\epsilon_{e} = 1/(2 \sigma^{R})
\]  
(2-27)

where \( \sigma_{y} \) is the original yielding stress.

At this moment, the equation (2-24) is the original yielding equation for a homogeneous isotropic material.

Generally, the total strain increments are classified into elastic strain increments and plastic strain increments. i.e.
\[
[\delta \epsilon] = [\delta \epsilon]^{e} + [\delta \epsilon]^{p}
\]  
(2-28)

According to the equation (2-24), the unloading criterion can be expressed as
\[
f(\sigma_{ij}, h^{'}) < 0 \text{ or } d\delta < 0
\]  
(2-29)

At this moment, we have
\[
[\delta \epsilon]^{e} = \{5\} [d\delta]^{e}
\]  
(2-30)

and
\[
[\delta \epsilon]^{p} = 0
\]  
(2-31)

in which
\[
\{5\} = \{0\}^{t}
\]

And the loading criterion can be expressed as
\[
f(\sigma_{ij}, h^{'}) = 0 \text{ or } d\delta > 0
\]  
(2-32)

At this moment, elastic strain increments \([\delta \epsilon]^{e}\) are still determined by (2-30), while plastic strain increments \([\delta \epsilon]^{p}\) are determined can be referred to the Ref [12].

D. TREATMENT OF PLANE PROBLEMS

A material-nonlinear finite element program of the quasi-three dimension problem developed by the authors can be used to analyze plane problems. Displacement expressions for a plane strain problem are
\[
\begin{align*}
  u &= 0 \\
  v &= V(y,z) \\
  w &= W(y,z)
\end{align*}
\]  
(2-33)

A plane strain problem, are substituted by new appropriate constants. These equations will become the equilibrium equations of a plane stress problem. Thus, a quasi-three dimension finite element program can be used to analyze not only a quasi-three dimension problem, if the appropriate limitations to boundary displacement conditions of a quasi-three dimension problem are made, but also a plane problem.

III. NUMERICAL SIMULATION OF STABLE PLANE CRACK GROWTH

Here, the material-nonlinear finite element program developed by the authors will be used to simulate the stable crack growth process of a plane crack. A rectangular plate made of 2024-T3 aluminum alloy with a centered line crack is subjected to monotonically and slowly increased loading up to the onset of fast fracture. The geometric configuration and the loading condition of the center-cracked specimen is shown in Fig.2.

The geometrical parameters, i.e., length 2L, width 2W, thickness B, initial crack size 2a, are listed as follows:
\[
\begin{align*}
  2L &= 27\text{ in.} \\
  2W &= 12\text{ in.} \\
  B &= 0.062\text{ in.} \\
  2a &= 6\text{ in.}
\end{align*}
\]  
(3-1)

The material property parameters, i.e., Young's modulus, Poisson's ratio, yield stress and linear hardening modulus are
\[
\begin{align*}
  E &= 10324\text{ ksi} \\
  v &= 0.33 \\
  \sigma_{y} &= 54.20\text{ ksi} \\
  H^{``} &= 259.74\text{ ksi}
\end{align*}
\]  
(3-2)

The finite element mesh and two paths for the evaluation of J-integral are referred to Ref [12]. Here, the load-crack size curve measured by experiment, as shown in Fig.3(a), is taken as the input data which is used the governing equation. The comparison of the experimental load-displacement curve with the numerical load-displacement data shown in Fig.3(b) proved that the numerical results here are correct.

A. J-INTEGRAL

Fig.4 showed that J-integral varied with crack size. It can be seen that the values of J-integral along the two paths are almost the same with the maximum difference being ±3.7%. This result was in agreement with that reported in [1.3]. Refs [1.3] pointed out that J-integral is still conservative in the condition of small amount of crack growth. Ref [11] reported that J-integral of far field for a compact tension specimen is in agreement with integral of near field in the condition of \( (a_{0}-a_{0})/(W-a_{0})<0.06 \). Here, \( (a_{0}-a_{0})/(W-a_{0})=0.03 \). Moreover, it can be seen that J-integral is proportional to the amount of crack growth in the origin of the stable crack growth. This result proved the two parameter characterization of fracture toughness properties proposed by Shih et al [11].

B. CRACK OPENING ANGLE

The two definitions of crack opening angle are one, the average crack opening angle, denoted by COA.
or $\sigma$, which is defined as the ratio of the crack opening displacement at the original crack tip to the total amount of crack growth that has occurred, while the other is the crack tip opening angle, denoted by CTOA or $\sigma$, which is defined as the ratio between crack opening displacement at a short and fixed distance $\Delta$ behind the current crack tip and $\sigma$.

In this work, $\Delta$ is taken to be the spacing between nodes located along the path of crack growth. The results of COA and CTOA are plotted in Fig. 5. It can be seen that COA and CTOA are both varied with the stable crack growth, but the range of the variation of CTOA is smaller than that of COA. From numerical values, the data of COA and CTOA reported here were almost the same as those reported in Ref [7].

C. CRACK TIP FORCE

The technique of releasing the crack tip forces is often used in the numerical simulation of stable crack growth. The result of crack tip force $F_c$ is shown in Fig. 6. It can be seen that $F_c$ is basically constant with the stable crack growth. This result proved the mixed criterion of $J_{21}$ and $F_c$ proposed by Kannine, et al.[2].

IV. CONCLUSION

In this paper, a material-nonlinear finite element program of the quasi-three dimension problem developed by the authors is described from a few respects. Further, it is illustrated how this program is used to analyze plane problems. Finally, a finite element analysis of the process of slow crack growth is made for a center-cracked specimen subjected to monotonically increased load until the point of fast fracture is reached. Numerical results and experimental results are compared. Variation laws of some fracture mechanics parameters are given with the stable crack growth.

REFERENCES

Fig. 3 Comparison of experimental load-displacement curve with numerical load-displacement data.

Fig. 4 The comparison of J-integral along different integral paths.

Fig. 5 The variations of CT01a and CT01b with the stable crack growth.

Fig. 6 The variation of the crack tip node force Fc with the stable crack growth.
HAMILTON'S PRINCIPLE AND THE EQUATIONS OF MOTION OF AN ELASTIC SHELL WITH AND WITHOUT FLUID LOADING

Cleon E. Dean and Michael F. Werby
Naval Oceanographic and Atmospheric Research Laboratory
CODE 221, Building 1100
Stennis Space Center, MS 39529-5004

Abstract—It has proven quite difficult to employ exact elastodynamic theory to describe the behavior of elastic vibrations on arbitrary bounded shells. In addition, exact theories preclude direct interpretation of particular features observed due to the excitation of elastic shell surfaces. A rather interesting approach to describe surface vibrations may be obtained by constructing a Hamiltonian in some approximate form that assumes some correlation of motion of the outer and inner shell surface. The class of theories that allow for this approach are referred to in applied mechanics as shell theories. The interesting feature of this Hamiltonian approach is that one can add various physical mechanisms to the Hamiltonian such as extensional motion, rotary inertia, shear distortion, fluid loading, etc., and thereby study the individual contributions to resonance patterns while adding physical insight to the fundamental processes that occur on shell surfaces. We develop shell theories in this manner and examine various contributions via Hamilton's principle. We believe that fluid loading has by and large not been treated adequately in the past, and we place particular emphasis on the treatment of that contribution to this work.

I. INTRODUCTION

The usual assumptions in shell theory (due to A. E. H. Love2) are:
(1) The thickness of a shell is small compared with the smallest radius of curvature of the shell; (2) The displacement is small in comparison with the shell thickness; (3) The transverse normal stress acting on planes parallel to the shell middle surface is negligible; (4) Fibers of the shell normal to the shell thickness; (3)-The transverse normal stress acting on planes parallel to the curvature of the shell;

We shall use these assumptions in the development of a shell theory in the style of a Timoshenko-Mindlin type theory. We wish to thank the Office of Naval Research, the Office of Naval Technology, and NOARL Management including Drs. Chin-Bing, Franchi, and Moseley for support of this work. Dr. Dean is at NOARL on an ONT Fellowship. This work was funded by NOARL Program Element 61153N.

The Lagrangian is the energy as applied to a spherical shell:

\[ L = T - V + W, \]

where the kinetic energy is

\[ T = \frac{1}{2} \rho \int_0^R \int_0^{2\pi} \int_0^\alpha \left( \dot{u}^2 + \dot{w}^2 \right) (\alpha + \beta)^2 \sin \theta d\theta d\phi, \]

with the surface displacements assumed linear in Timoshenko-Mindlin fashion:

\[ \dot{u}_s = \left( 1 + \frac{x}{\alpha} \right) \dot{u} - \frac{\alpha}{\alpha + \beta} \frac{\partial \dot{w}}{\partial \theta}, \]

\[ \dot{w}_s = \frac{\partial \dot{w}}{\partial \phi}. \]

There is no movement in the v-direction since the restraining field can be taken to be torsionless. By substitution, the kinetic energy becomes

\[ T = \pi \rho \int_0^R \int_0^{2\pi} \int_0^\alpha \sin \theta \left( \frac{h_1^2}{8\alpha^2} + \frac{h_2^2}{8\alpha^2} - 2 \left( \frac{h_1^2}{8\alpha^2} + \frac{h_2^2}{8\alpha^2} \right) \frac{\partial \dot{u}}{\partial \theta} \right) \frac{\partial \dot{w}}{\partial \phi} \]

+ \left( \frac{h_1^2}{8\alpha^2} + \frac{h_2^2}{12} \right) \frac{\partial \dot{w}}{\partial \theta}^2 + \left( \frac{h_3^2}{12} \right) \dot{w}_s^2 d\theta, \]

or finally,

\[ T = \pi \rho \int_0^R \int_0^{2\pi} \int_0^\alpha \sin \theta \left( \frac{h_1^2}{8\alpha^2} + \frac{h_2^2}{8\alpha^2} + 1 \right)^2 - \left( 3.6 \beta_1^2 + 6 \beta_2^2 \right) \frac{\partial \dot{u}}{\partial \theta} \frac{\partial \dot{w}}{\partial \phi} \]

+ \left( 1.8 \beta_1^2 + \beta_2^2 \right) \frac{\partial \dot{w}}{\partial \theta}^2 + \left( \beta_2^2 + \beta_2^2 \right) \dot{w}_s^2 \sin \theta d\theta d\phi. \]

II. DERIVATION OF EQUATIONS OF MOTION

For spherical shells, membrane stresses (proportional to \( \beta \)) predominate over flexural stresses (proportional to \( \beta^2 \)) where

\[ \beta = \frac{1}{\sqrt{12}} \frac{h}{a}. \]

We differ from the standard derivation for the sphere (due to Janer and Feit1) by retaining all terms of order \( \beta^2 \) in both the kinetic and potential energy parts of the Lagrangian. We note that this level approximation will allow us to include the effects of rotary inertia and shear distortion in our shell theory, as well as the usual extensional motion of the shell. The new derivation is as follows: let \( a, v, w \) axis system be set up on a spherical shell of radius \( a \) (measured to shell thickness) with thickness \( h \), as shown in Fig. 1.

Then the new Lagrangian (which is equivalent to a Timoshenko-Mindlin theory as applied to a spherical shell) is

\[ L = T - V + W \]

In a similar fashion the potential energy is

\[ V = \frac{1}{2} \int_0^R \int_0^{2\pi} \int_0^\alpha \left( \sigma_{xx} \dot{w}^2 + \sigma_{yy} \dot{w}^2 \right) (\alpha + \beta)^2 \sin \theta d\theta d\phi, \]

\[ = \frac{1}{2} \int \int \frac{1}{2} E \left( \frac{\partial \dot{u}}{\partial \theta} - \frac{\alpha}{\alpha + \beta} \frac{\partial \dot{w}}{\partial \phi} \right)^2 + \left( (1 + \frac{x}{\alpha} \frac{\partial u}{\partial \theta} - \frac{\alpha}{\alpha + \beta} \frac{\partial \dot{w}}{\partial \phi} \right)^2 \]

\[ + \left( \frac{x}{\alpha} \frac{\partial \dot{w}}{\partial \phi} \right)^2 \sin \theta d\theta d\phi. \]
where the nonvanishing strain components are
\[ \varepsilon_{\theta\theta} = \frac{1}{a} \left( \frac{\partial u}{\partial \theta} + w \right) + \frac{x}{a^2} \frac{\partial u}{\partial \theta} \] (7a)
and
\[ \varepsilon_{ww} = \frac{1}{a} \left( \frac{\partial w}{\partial \theta} + u \right) + \frac{x}{a^2} \frac{\partial w}{\partial \theta} \] (7b)

The nonzero stress components are
\[ \sigma_{\theta\theta} = \frac{E}{1-\nu^2} (\varepsilon_{\theta\theta} + \nu \varepsilon_{ww}), \quad \sigma_{ww} = \frac{E}{1-\nu^2} (\varepsilon_{ww} + \nu \varepsilon_{\theta\theta}). \] (8a,b)
The work done by the surrounding fluid on the sphere is
\[ W = 2\pi a^2 \rho_f \bar{w} \sin \theta d\theta. \] (9)

Since the integration along the polar angle \( \theta \) is intrinsic to the problem, the solution must use a Lagrangian density:
\[ L_1 = \pi \rho_f a^4 [(1 + \beta^2)u^2 - 6\beta^2 \frac{\partial \bar{u}}{\partial \theta} + \beta^2 \frac{\partial \bar{u}}{\partial \theta}^2 + (v + \frac{\partial \bar{u}}{\partial \theta} + (u + v \cot \theta)^2 \right]
\[ +2v \left( \frac{\partial \bar{w}}{\partial \theta} + u \right)^2 + 2v \cot \theta \left( \frac{\partial \bar{w}}{\partial \theta} + u \right)^2 \right] \sin \theta + 2\pi a^2 \rho_f \bar{w} \sin \theta, \] (10)

with differential equations
\[ \frac{\partial \bar{u}}{\partial \theta} - \frac{d}{d\theta} \frac{\partial \bar{u}}{\partial \theta} = 0, \] (11a)
and
\[ \frac{\partial \bar{w}}{\partial \theta} - \frac{d}{d\theta} \frac{\partial \bar{w}}{\partial \theta} = 0. \] (11b)

By substitution we have
\[ (1 + \beta^2) \left[ \frac{\partial \bar{u}}{\partial \theta} + \cot \theta \frac{\partial \bar{u}}{\partial \theta} - (v + \cot^2 \theta) \right] - \beta^2 \frac{\partial \bar{w}}{\partial \theta} \]
\[ - \beta^2 \cot \theta \frac{\partial \bar{w}}{\partial \theta}^2 + [(1 + v) + \beta^2 (\cot^2 \theta)] \frac{\partial \bar{w}}{\partial \theta} \]
\[ - \frac{a^2}{c_p^2} [(1 + 6\beta^2) \bar{u} - 3\beta^2 \frac{\partial \bar{w}}{\partial \theta}] = 0, \] (12a)

and
\[ \beta^2 \frac{\partial \bar{u}}{\partial \theta}^2 + 2 \beta^2 \cot \theta \frac{\partial \bar{u}}{\partial \theta}^2 - [(1 + v)(1 + \beta^2) + \beta^2 \cot^2 \theta] \frac{\partial \bar{u}}{\partial \theta} \]
\[ + \cot \theta (2v + \cot^2 \theta) \beta^2 (1 + \beta^2) - \beta^2 \frac{\partial \bar{w}}{\partial \theta}^2 - 2 \beta^2 \cot \theta \frac{\partial \bar{w}}{\partial \theta}^2 \]
\[ + \beta^2 (1 + v + \cot^2 \theta) \beta^2 \cot \theta (2v + \cot^2 \theta) \frac{\partial \bar{w}}{\partial \theta} \]
\[ - 2(1 + v) \bar{w} \frac{a^2}{c_p^2} (1 + \beta^2) = -p_a \left( (1 - v^2) \frac{a^2}{Eh} \right). \] (12b)

These differential equations have solutions of the form
\[ u(\eta) = \sum_{n=0}^{\infty} U_n (1 - \eta^2)^{\frac{1}{2}} \frac{dP_n}{d\eta}, \quad w(\eta) = \sum_{n=0}^{\infty} W_n P_n (\eta), \] (13a,b)
where \( \eta = \cos \theta \) and \( P_n (\eta) \) are the Legendre polynomials of the first kind of order \( n \). The differential equations of motion (12a,b) are satisfied if the expansion coefficients \( U_n \) and \( W_n \) satisfy a homogeneous system of linear equations whose determinant yields a frequency equation of the form
\[ 0 = \Omega^2 (1 + f_2) - \Omega^2 [(1 + \beta^2) + (1 + f_2) + (1 + v) + \beta^2 \lambda_2] \]
\[ + [(1 + v) + \beta^2 \lambda_2] + \beta^2 \lambda_2 \] (14)

where \( \lambda = v + \lambda_2 - 1 \) and \( \lambda_2 = n(n + 1) \). The fluid loading is including by the term \( f_2 \), which uses the exact volume equation of a sphere to include the influence of the exterior spherical shell of fluid that influences the motion of the vibrating spherical shell. This is done in a manner analogous to that for the fluid loaded plate: for a plate, fluid loaded on one side, of mass per-unit area \( p_h \), the appropriate nondimensional measure of fluid-loading at any frequency \( \nu \) is \( p_c \nu \). The surface pressure for a sphere in terms of modal specific acoustic impedances \( z_a \) is
\[ p(\alpha, \theta, \phi) = \sum_{n=0}^{\infty} r_n \hat{W}_n P_n (\cos \theta) \cos \phi, \] (15)

where
\[ \lambda_n = \nu \rho_c \hat{k}_n (ka). \] (16a)

Let
\[ \lambda_n = -i \omega_n. \] (16b)

Then
\[ r_n = \nu \rho_c \hat{k}_n (ka). \] (17a)

Similarly
\[ m_n = \nu \rho_c \hat{k}_n (ka). \] (17b)

The fluctuating pressure on the surface of the sphere constitutes the radiation loading. In Zanger and Feit's derivation of the fluid loading,
\[ f_a = \frac{m_n}{\rho_c \lambda}, \] (18)
but we use
\[ f_a = \frac{3a^2 m_n}{\rho_c (a^2 - (a - h)^2)}. \] (19)

which has the advantage of an exact volume difference for the amount of fluid affecting the vibrating shell.

III. CONCLUSIONS

We expect this new derivation of the equations of motion for a spherical shell to give new insight into the types of waves supported by a fluid loaded spherical or spheroidal shell. From previous work\(^4\) we expect the effects of fluid loading to be more important for antisymmetric than for symmetric modes of vibration, and for our improved volume measure to be more significant for thicker shells. We also hope to see an improvement in the asymptotic behavior of the resonances in the high size parameter limit for the antisymmetric case. That is, we expect the results to go to the Rayleigh velocity limit as the relative radius of the shell goes to infinity.

Abstract.- In this paper, a dynamic thermoelastic model for linear beams is obtained by asymptotic analysis. The model is obtained as the weak limit of a rescaled three-dimensional model after a change of variable to a fixed domain. As well as, a compatibility condition in order to obtain strong convergence is given.

I. SETTING OF THE PROBLEM

The basic ideas of the asymptotic expansion method introduced by BERMUDEZ – VIANO [2] and TRABUCHO – VIAÑO [8,9] are applied here in order to obtain the justification of an evolution thermoelastic model for linearized rods. We study the case of an elastic beam \( \Omega^\epsilon = \omega^\epsilon \times (0,L) \), \( \epsilon^2 = \text{meas}(\omega^\epsilon) \), clamped at both ends \( \Gamma_0^\epsilon = \omega^\epsilon \times \{0,L\} \) and kept at them to an uniform reference temperature \( T_0 \). The beam is supposed to be submitted to body forces \( F^\epsilon(x) \) in \( \Omega^\epsilon \) and surface forces \( \sigma^\epsilon(x) \) on \( \Gamma_0^\epsilon = \omega^\epsilon \times \{0,L\} \) and initial conditions in displacement, velocity and temperature \((\bar{U}^\epsilon, \dot{V}^\epsilon, T^\epsilon)\). We assume the beam \( \Omega^\epsilon \) made of an isotropic, homogeneous, elastic material of Saint Venant – Kirchhoff, with Young's modulus \( \epsilon \), Poisson's ratio \( \nu \), thermal dilation coefficient \( \alpha \), heat conductivity coefficient \( k \) and specific heat coefficient \( C \), independent of \( \epsilon \). We denote by \( \rho^\epsilon \) the mass density and we assume that \( \rho^\epsilon = \epsilon^2 \rho^0 \) with \( \rho \) independent of \( \epsilon \) [3,7]. Then, the dynamic problem corresponding to the thermoelastic behavior of the beam \( \Omega^\epsilon \) during the time interval \( [0,T] \) is governed by the classical evolutionary system of equations posed as a function of the displacement field \( \bar{U}^\epsilon \) and the temperature increment field \( T^\epsilon = T^\epsilon - T_0 \) ([5])

\[
\rho^\epsilon \ddot{U}^\epsilon - \sigma^\epsilon + \Gamma^\epsilon = F^\epsilon \quad \text{in} \quad \Omega^\epsilon \times (0,T), \quad (1)
\]

\[
\rho^\epsilon \ddot{U}^\epsilon + \frac{1}{2} \sigma^\epsilon (k \sigma^\epsilon + \sigma^\epsilon) - \frac{E \alpha}{1-2\nu} \epsilon_k^\epsilon (U^\epsilon) \text{in} \quad \Omega^\epsilon \times (0,T), \quad (2)
\]

\[
U^\epsilon = 0, \quad \sigma^\epsilon = 0 \quad \text{on} \quad \Gamma_0^\epsilon \times (0,T), \quad (3)
\]

\[
\Sigma_k \; n_k = G_k^\epsilon, \quad k \delta^\epsilon \sigma^\epsilon \; n_k = 0 \quad \text{on} \quad \Gamma_0^\epsilon \times (0,T), \quad (4)
\]

\[
U^\epsilon (0) = \bar{U}^\epsilon, \quad U^\epsilon (T) = \bar{U}^\epsilon, \quad \sigma^\epsilon (0) = \bar{\sigma}^\epsilon, \quad \sigma^\epsilon (T) = \bar{\sigma}^\epsilon. \quad (5)
\]

In (1)–(5) the Piola – Kirchhoff tensor \( \Sigma^\epsilon \) is given by the linear thermoelastic law:

\[
e^\epsilon (U^\epsilon) = \alpha \Theta^\epsilon I = \frac{1+y^\epsilon}{E} \Sigma^\epsilon - \frac{\nu^\epsilon}{E} \text{tr}(\Sigma^\epsilon) I, \quad (6)
\]

where \( e_{ij}^\epsilon (U^\epsilon) = \frac{1}{2} (\sigma_{ij}^\epsilon U_{ij}^\epsilon + \sigma_{ij}^\epsilon U_{ij}^\epsilon) \).

In a classical way the existence and uniqueness of solution of the three-dimensional problem (1)–(5) with the following regularity, is obtained (6):

\[
U^\epsilon \in C^0(0,T; \Omega^\epsilon) \cap C^1(0,T; \Omega^\epsilon) \cap C^2(0,T; \Omega^\epsilon) \cap C^3(0,T; \Omega^\epsilon), \quad (6)
\]

\[
\sigma^\epsilon \in C^0(0,T; \Omega^\epsilon) \cap C^1(0,T; \Omega^\epsilon) \cap C^2(0,T; \Omega^\epsilon) \cap C^3(0,T; \Omega^\epsilon).
\]

II. WEAK CONVERGENCE

Following the works by BLANCHARD – FRANCÉT [3], BERMUDEZ – VIANO [2] and TRABUCHO – VIAÑO [8,9] we study the behavior of the three-dimensional model as the area of the cross-section of the beam tends to zero. In order to make possible we consider a variational formulation of the problem. Then, we apply the following change of variable and scale, which allows us to work on the fixed domain \( \Omega = \omega \times (0,L), \omega = \epsilon^{-1} \omega^\epsilon \):

\[
\pi^\epsilon : x=(x_1, x_2, x_3) \in \Omega \rightarrow \pi^\epsilon(x) = x^\epsilon = (\epsilon x_1, \epsilon x_2, x_3) \in \Omega^\epsilon, \quad (7)
\]

\[
u^\epsilon = \epsilon^2 \nu^0 \quad \text{with} \quad \nu \in [0,1], \quad (8)
\]

\[
\sigma_{ij}^\epsilon = \epsilon^2 \sigma_{ij}^0, \quad e_{ij}^\epsilon (U^\epsilon) = \epsilon^2 e_{ij}^0 (U^\epsilon) \quad \text{weakly in} \quad H^1(0,T; \Omega^\epsilon). \quad (9)
\]

\[
é^\epsilon : \epsilon \rightarrow 0 \quad \text{weakly} \quad \text{in} \quad H^1(0,T; \Omega^\epsilon). \quad (10)
\]

\[
\Gamma^\epsilon : \epsilon \rightarrow 0 \quad \text{weakly} \quad \text{in} \quad L^2(\Omega). \quad (11)
\]

Then, for the sequence \( (u^\epsilon, v^\epsilon, \sigma^\epsilon) \) and, at least, for a subsequence of \( (\epsilon, \nu(\epsilon), \sigma(\epsilon)) \), noted in the same way, we have:

888
\[ u(c) \to u^0 \quad \text{weak-* in } L^\infty(0,T; H), \]
\[ u_\alpha(c) \to u^0_\alpha, \quad \text{as } u_\alpha(c) \to 0, \quad \epsilon^2 a_{33}(c) \to \sigma^0_{33}, \]
\[ \epsilon a_{33}(c) \to \sigma^0_{33}, \quad \sigma_{33}(c) \to \sigma^0_{33}. \]
\[ e^{-2} e_\alpha(u(c)) = \frac{1+\nu}{E} \sigma^0_{\alpha\alpha} \quad \text{weak-* in } L^\infty(0,T; L^2(\Omega)), \]
\[ \sigma(c) \to \sigma^0 \quad \text{weak-* in } L^\infty(0,T; L^2(\Omega)) \quad \text{and weakly in } L^2(0,T; W), \]
\[ e^{-1} a_\alpha(c) \to r^0_\alpha \quad \text{weakly} \quad \text{in } L^2(0,T; L^2(\Omega)). \]

where:

(i) \( u^0 \) is a Bernoulli-Navier field, \( u^0 = (u_1^0, u_2^0, u_3^0 - \kappa_e \partial_3 u_\alpha^0) \),
where its components are determined by:

a) \( u^0_\alpha \in L^\infty(0,T; H^0(\Omega,L)) \) is the unique solution of the problem

\[ g(\epsilon - 1) a_{33} u_\alpha^0 = f_\omega \sigma^0_{\alpha\alpha} + f_\gamma \sigma^0_{\alpha\gamma} + a_3 (f_\omega \sigma^0_{\alpha\gamma} + f_\gamma \sigma^0_{\alpha\gamma}) \]
\[ u^0_\alpha(0) = \sigma^0_{\alpha\alpha}(0) = f_\omega \sigma^0_{\alpha\alpha} - f_\omega \sigma^0_{\alpha\gamma}, \quad \epsilon \neq 0. \]

b) \( (u_3^0, q^0) \in L^\infty(0,T; H^0(\Omega)) \times L^\infty(0,T; H^0(\Omega)) \) is the unique solution of the coupled system:

\[ E \partial_{33} u_3^0 - E \kappa_e \partial_3 u_\alpha^0 = f_\omega \sigma^0_{\alpha\gamma} - f_\gamma \sigma^0_{\alpha\gamma}, \quad \partial_3 u_3^0 = \sigma^0_{\alpha\alpha} \quad \text{weakly in } L^2(0,T; L^2(\Omega)). \]

Finally, the same methods can be applied, with slight modifications, to study the behavior of the beam under different boundary conditions (beam clamped only at one end, heat flux over the surface of the beam...) and/or nonhomogeneous material.

This research was partially supported by European project SCI*0473-C (EDB) and by project PB87-0481 of DGICYT of Spain.

REFERENCES

A NUMERICAL APPROACH FOR A CLASS OF AXIALLY SYMMETRIC DEFORMATIONS IN NONLINEAR ELASTICITY

E.M. Croitoru and M.F. Pettigrew
Department of Applied Mathematics
The University of Western Ontario
London, Canada, N6A 5B9

Abstract This work is concerned with the deformation, the stress field and the stress concentration in a thick plate with a cylindrical hole or inclusion that is subject to a pressure at the opening while all other boundaries remain traction-free. The equations that model such a deformation are highly nonlinear and coupled. We seek a numerical solution and develop an algorithm based on a Newton-Kantorovich linearization. The equations are discretized with fourth order compact finite differences over a variable grid. The deformation, the stress field and the stress concentration are investigated throughout and are compared with results of linear elasticity theory.

A large class of elastomers, synthetics and some biological tissues obey the theory of Finite Elasticity. The set of partial differential equations that models such materials under stress are usually nonlinear and coupled and few exact analytical solutions are known so far [1],[2].

The problem we are concerned here belongs to a group of boundary value problems involving perturbations about large deformations in thick plates weakened by holes or inclusions. The presence of holes, notches and inclusions results in a stress concentration around the opening and these types of problems are of significant interest in design.

We consider firstly that the plate undergoes a large deformation due to a pressure applied at the opening. The body is further subject to some perturbative surface tractions, reaching a final state of equilibrium. Two types of problems arise: the problem where the thickness of the plate is prevented from changing - this is a two dimensional plane-strain problem - and the problem where the thickness is allowed to change and we deal with a three-dimensional case.

We found the general analytical solution [3] for a class of two-dimensional boundary value problems: it applies to incompressible materials with strain-energy function of Mooney-Rivlin type, although the method of solution is not restricted to this particular form of the strain-energy. A typical point located at \((r, \theta, z)\) in the unstrained and unstressed state moves to \((r, \theta, z)\) after the large deformation takes place. The body is then subject to a certain perturbation which is viewed as a perturbative displacement field of components \(u(r, \theta)\) and \(v(r, \theta)\) in the \(r\) and \(\theta\)-direction respectively. The small parameter \(\varepsilon<<1\) has its own physical significance in each specific problem. We have found all possible combinations of displacement fields \((u_0, v_0)\) that can be superimposed on the initial large deformation such that the equilibrium can be maintained without body forces, by surface tractions alone. The solution for \(u(r, \theta)\) and \(v(r, \theta)\) is expressed in terms of Fourier series where the coefficients themselves are functions of \(r\). The \(r\)-dependent functions, in turn, are solutions of a fourth order ordinary differential equation and are expressible in series form with logarithmic contribution.

The general solution is suitable for solving a large class of boundary value problems [3],[4]. A special attention is devoted to the case of a perturbative uniaxial tension applied at great distances from the opening. The section passing through the axis of the hole and which is perpendicular to the line of stretch is the most stressed one. If the body is subject solely to the perturbative uniaxial loading and no previous large deformations occur, the stress at the edge of the hole is three times the stress applied at infinity - a result obtained within the linear elasticity theory [5]. On the other hand, the body would assume a certain stress field if only a large deformation took place. Now, we consider that the uniaxial tension is applied on the existing finite deformation. We deal here with nonlinearity and we know that superposition doesn't hold. The stress concentration at the hole could be magnified or diminished. We have found a definite increase in stress concentration, as a nonlinear effect. However, the highly stressed region is very localized near the hole and the stress field approaches very rapidly the value prescribed at infinity.

The analysis for the three-dimensional case is more involved. Firstly, we consider that the layer is subject to a pressure applied at the opening, all the other boundaries remain traction-free. The deformation is axially symmetric and is given by the mapping

\[ \rho = f(r, \theta), \quad \theta = \theta, \quad \zeta = g(r, \theta). \]  

While the differential equation that describes the corresponding large deformation in the two-dimensional case can easily be integrated regardless of the functional form of the strain-energy function to obtain \(p = (r^2 - k)^{\alpha/2}\), the set of partial differential equations here are nonlinear and strongly coupled, and to find the analytical solution that satisfies completely and rigorously all the boundary conditions is a difficult task.

Some progress can be made by specifying the strain-energy function. We assume a Neo-Hookean material where \(W = C_1 r^3\).

The equilibrium equations take the form

\[ \frac{1}{2C} \frac{\partial p}{\partial r} \cdot \frac{\partial}{\partial r} \left\{ f^2 \left[ \frac{\partial^2 u}{\partial r^2} + \frac{\partial v}{\partial r} \right] \right\} - \frac{\partial}{\partial r} \left\{ f^2 \left[ \frac{\partial u}{\partial r} + \frac{\partial v}{\partial \theta} \right] \right\} = 0, \]

\[ \frac{1}{2C} \frac{\partial p}{\partial z} - \frac{\partial}{\partial r} \left\{ f^2 \left[ \frac{\partial^2 u}{\partial r^2} + \frac{\partial v}{\partial r} \right] \right\} + \frac{\partial}{\partial r} \left\{ f^2 \left[ \frac{\partial u}{\partial r} + \frac{\partial v}{\partial \theta} \right] \right\} = 0, \]

where \(p(r, \theta)\) is the hydrostatic pressure.

The incompressibility condition requires that

\[ \frac{f}{r} \left( \frac{\partial u}{\partial \theta} - \frac{\partial v}{\partial r} \right) = 1. \]
We have found a solution in terms of Bessel functions using a successive approximation method [6]. However, the solution is limited to moderately large deformations.

In order to cast further light on the three-dimensional stress concentration, we seek a numerical solution based on an iterative procedure between \( f(r,\theta) \), \( g(r,\theta) \) and \( p(r,\theta) \).

The formulation for \( f \), \( g \) and \( p \) involves two coupled second order equations in \( f \) and \( g \); it turns out that the nonlinearity appears in the first order terms only. The computational aspects are considerably simplified by introducing the following two transformations:

- the first transformation
  \[
  x = \frac{a}{r}, \quad y = \frac{\xi}{l},
  \]
  where \( a \) is the radius of the cylinder and \( 2l \) is the plate thickness.

- the second transformation given by
  \[
  F(xy) = \frac{\xi}{a} f(p,\xi), \quad G(xy) = \frac{1}{l} g(p,\xi),
  \]
  removes the leading asymptotic behaviour in \( F \).

The domain of computation becomes

\[
D : \{ (x,y) \mid 0 \leq x \leq 1, \quad 0 \leq y \leq 1 \}.
\]

The resulting set of nonlinear equations and the boundary conditions involve the aspect ratio \( \gamma = a/l \) which we take as a parameter.

Denote the \( k \)-th approximate solution by \( \Psi_k = [F_k, G_k, P_k] \).

We impose a rectangular grid over \( D \) with mesh refinement in the vicinity of the opening to accommodate the localized stress concentration. We apply a Newton-Kantorovitch linearization [7],[8] and discretize the linearized equations with variable grid fourth order compact finite differences [9]. The resulting linear algebraic system of equations turns out to be sparse. An iterative sparse matrix solver [10] allows us to obtain \( F(x,y) \) and \( G(x,y) \). As some terms in the expression for \( P(xy) \) become singular at \( x=0 \), we introduce a new function \( \Psi(x,y) \) such that

\[
\frac{\partial \Psi}{\partial x} = \frac{\partial \Psi}{\partial y}, \quad \frac{\partial \Psi}{\partial y} = -\frac{\partial \Psi}{\partial x},
\]

and hence \( \nabla \Psi = 0 \) in \( D \). From the computation of \( \Psi(x,y) \) new values of the hydrostatic pressure and its gradient are obtained. This completes the iteration taking \( \Psi \) into \( \Psi^{k+1} \). The deformation, the stress field and the stress concentration around the opening is investigated in detail. The computational results are presented graphically and nonlinear effects are highlighted.

\textbf{Acknowledgements} This work has been partially supported by funds provided by the Natural Sciences and Engineering Research Council of Canada.

\textbf{References}

ORTHOGONAL WITH NON-INTEGRABLE WEIGHT FUNCTION JACOBI POLYNOMIALS
AND THEIR APPLICATION TO SINGULAR INTEGRAL EQUATIONS
IN ELASTICITY AND HEAT CONDUCTION PROBLEMS

Igor PODLUBNY
Department of Control Engineering
Faculty of Mining, Technical University
B.Nemcovej 3, 04200 Kosice, Czechoslovakia

Abstract—Two new regularization formulae for evaluating the finite part of the integrals with non-integrable Jacobi weight function and new spectral relationships for orthogonal with non-integrable weight Jacobi polynomials are obtained. A new approach for solving singular integral equations with Cauchy's kernel, based on the obtained spectral relationships and regularization formulae, is proposed. Such approach can be used for the characteristic singular equation as well as for the complete one.

I. INTRODUCTION: THE SHORTEST HISTORY OF THE PROBLEM

1) The definition of the finite part of a divergent integral was given by Hadamard [1], when he was considering the integral

\[
I(\lambda) = \int f(x) x^{-\lambda} dx
\]

for \( \lambda < -1 \). He obtained and used first regularization formulae for the divergent integrals with the non-integrable weight function.

2) The authors of the Bateman's project [2] noticed that the majority of the relationships for Jacobi polynomials can be used even when \( \alpha < -1 \) or \( \beta < -1 \) or both \( \alpha < -1 \) and \( \beta < -1 \).

3) First application of such Jacobi polynomials was found by Popov and Onischuk [3]. They reduced a problem for a plate with a rigid inclusion to the integral equation with so called smooth kernel, for which Jacobi polynomials can be used even when \( \alpha < -1 \) or \( \beta < -1 \), as they proved, are the eigenfunctions. It allowed them to obtain the solution of the integral equation in the form of Fourier-Jacobi series.

II. REGULARIZATION FORMULAE

\textbf{Lemma 1.} If \( f(x) \in C^1[-1,1], \alpha < 1, \beta < 1 \), then

\[
\int_{-1}^{1} \frac{f(t)P_{n}^{-\alpha-1,-\beta-1}(t)dt}{(1-t)^{\alpha+1}{(1+t)}^{\beta+1}} = \int_{-1}^{1} \frac{f'(t)P_{n-1}^{-\alpha,-\beta}(t)dt}{(1-t)^{\alpha+1}{(1+t)}^{\beta+1}} - \frac{1}{2n}\int_{-1}^{1} \frac{f'(t)P_{n}^{-\alpha,-\beta}(t)dt}{(1-t)^{\alpha+1}{(1+t)}^{\beta+1}}
\]

for \( n = 1, \omega \)

\textbf{Lemma 2.} If \( f(x) \in C^1[-1,1], \alpha < 1, \beta < 1 \), then

\[
\int_{-1}^{1} \frac{f(t)dt}{(1-t)^{\alpha+1}{(1+t)}^{\beta+1}} = \frac{1}{4} \int_{-1}^{1} \frac{f'(t)dt}{(1-t)^{\alpha+1}{(1+t)}^{\beta+1}} + \frac{1}{4} \int_{-1}^{1} \frac{f'(t)dt}{(1-t)^{\alpha+1}{(1+t)}^{\beta+1}}
\]

Some particular cases of Lemma 2 are listed below:

\[
\int_{-1}^{1} \frac{f(t)dt}{(1-t)^{\alpha+1}{(1+t)}^{\beta+1}} = \frac{1}{4} \int_{-1}^{1} \frac{f'(t)dt}{(1-t)^{\alpha+1}{(1+t)}^{\beta+1}} + \frac{1}{4} \int_{-1}^{1} \frac{f'(t)dt}{(1-t)^{\alpha+1}{(1+t)}^{\beta+1}}
\]

\[
\int_{-1}^{1} \frac{f(t)dx}{(1-x)^{\alpha+1}{(1+x)}^{\beta+1}} = \frac{1}{4} \int_{-1}^{1} \frac{f'(t)dx}{(1-x)^{\alpha+1}{(1+x)}^{\beta+1}} + \frac{1}{4} \int_{-1}^{1} \frac{f'(t)dx}{(1-x)^{\alpha+1}{(1+x)}^{\beta+1}}
\]

\[
\int_{-1}^{1} \frac{f(t)dx}{(1-x)^{\alpha+1}{(1+x)}^{\beta+1}} = \frac{1}{4} \int_{-1}^{1} \frac{f'(t)dx}{(1-x)^{\alpha+1}{(1+x)}^{\beta+1}} + \frac{1}{4} \int_{-1}^{1} \frac{f'(t)dx}{(1-x)^{\alpha+1}{(1+x)}^{\beta+1}}
\]

III. SPECTRAL RELATIONSHIPS

\textbf{Lemma 3.} If \( \alpha \) is not integer, then

\[
\int_{-1}^{1} \frac{q_{m+\alpha}^{n}(x)\varphi(x)dx}{(1-x)^{\alpha+1}{(1+x)}^{\beta+1}} = \int_{-1}^{1} \frac{q_{m+\alpha}^{n}(x)\varphi(x)dx}{(1-x)^{\alpha+1}{(1+x)}^{\beta+1}}
\]

where \( q_{m+\alpha}^{n}(\tau) = \frac{\tau^{m+\alpha}P_{n+m}\sin(\alpha \pi \tau)}{\sin(\alpha \pi \tau)} \),

\[
q_{m+\alpha}^{n}(\tau) = \frac{\tau^{m+\alpha}P_{n+m}\sin(\alpha \pi \tau)}{\sin(\alpha \pi \tau)}
\]

for \( m=0, \omega \) and \( k=0, \omega \) when \( m+n+k \geq 0 \).

892
Lemma 4. If $\alpha$ is not integer, when
\[ 1 - \alpha + k \begin{pmatrix} (1-t)^{\alpha+m} \end{pmatrix} dt = \frac{t^{-\alpha+k} - (1-t)^{-\alpha+k}}{\tau - t} \]
\[ = \frac{(-1)^{k+1} \alpha-k, \alpha-m}{m+k} = \sin(\alpha \cdot \pi) \]
where $m+k \geq 0$.

IV. THE SOLUTION OF THE CHARACTERISTIC SINGULAR INTEGRAL EQUATION

The use of the previous results is illustrated on the example of the characteristic singular integral equation of the first kind:

\[ f(t) = \int_{-1}^{1} \frac{f(x) \sin(\alpha x)}{t-x} dx \]  

Using [4], we look for the solution in the following form:

\[ f(t) = \sum_{n=0}^{\infty} \left( A_n w_1(t) + (1-t)^{3/2} \right) \left( \sum_{n=0}^{\infty} \rho_n p_n(x) \right) \]

The constants $\rho_n$ are obtained independently on the class of solutions are given by the following expression:

\[ \rho_n = 8 \rho_{n+3} + \int \frac{f(x) \sin(\alpha x)}{t-x} \sin(\alpha x) dx \]

The constants $A_n$ are independent on the class of solutions:

\[ A = \frac{1}{\pi} \int_{-1}^{1} f(t) dt = 1 \]

 garanting the unique solution we obtain:

\[ A_1 = \frac{1}{2} f_0 - f_1 - f_2 = \frac{3f_0}{2} - \frac{3f_3}{2} \]

\[ A_2 = \frac{1}{4} f_0 - f_1 - f_2 = \frac{3f_0}{4} - \frac{3f_3}{4} - \frac{1}{2} \]

\[ A_3 = \frac{1}{8} f_0 - f_1 - f_2 = \frac{3f_0}{8} - \frac{3f_3}{8} - \frac{1}{4} \]

b) For the solution unbounded at the left end and bounded at the right end of $(-1,1)$ we obtain:

\[ A_1 = 0, \quad A_2 = c_0, \quad A_3 = \frac{3f_3}{4} - \frac{1}{2} \]

V. CONCLUSIVE REMARKS

The proposed approach can be used also for solving characteristic singular equations of the second kind and for solving complete singular equations of the first and the second kinds with Cauchy’s kernel. In the comparison with the famous method the proposed one has some advantages which are stipulated by the common form of the solution (2).

Finally, if $f(x) \in C[-1,1]$, then the series (2) uniformly converges on $[-1+\epsilon, 1-\epsilon]$, $0 < \epsilon < 1$, and $n \to \infty$.

REFERENCES:


FINITE DIFFERENCE SIMULATION OF TIME-DEPENDENT FLOW AROUND A LONGITUDINALLY OSCILLATING CYLINDER

PAPOLU MANIKYALA RAO
Prof.M. Kawahara Laboratory, Department of Civil Engg, CHUO University, Kusagi, Fukuoka, Japan.

AND

KUNIO KUWAHARA
The Institute of Space and Astronautical Science, Sagamihara-Shi, Kanagawa, 229, Japan

ABSTRACT- A finite difference solution is obtained for the time dependent viscous incompressible 2-dimensional flow past a longitudinally oscillating circular cylinder. A Navier-Stokes equation of finite difference form is solved by moving grid system, based on a time-dependent coordinate transformation. The solution describes the development of the vortex street developed behind the cylinder when cylinder remains stationary. The time dependent lift and drag coefficients are obtained. The power spectra of the drag, lift and displacement is also reported. The computer results predict the lock-in phenomenon which occurs when oscillation frequency is around double the natural vortex shedding frequency.

INTRODUCTION

The understanding of the characteristics of vibrations induced by vortex shedding is of great importance in the design of structures such as heat exchangers, offshore platforms, power cables, etc. The determination of the forces acting on a cylinder undergoing harmonic in-line oscillations is of special interest in aerelasticity as well as for a basic understanding of the fluid mechanics. Information about the in-line oscillations of a cylinder in uniform flow has not been studied extensively. However, only a few studies have been made for a circular cylinder oscillating in uniform flow[1-4].

The numerical simulation of steady flow past a circular cylinder undergoing in-line and/or transverse oscillations through the use of two-dimensional unsteady Navier-Stokes equations was taken by [5] for relatively small amplitudes. Recently, numerical solutions of unsteady flow about stationary and oscillating cylinders have been obtained[6-9].

THE GOVERNING EQUATIONS AND NUMERICAL METHOD

The non-dimensional governing Navier-Stokes equations, written in non-conservative form and expressed in the dimensionless quantities, are

\[ \frac{\partial V}{\partial t} + (V \cdot \nabla)V = - \nabla p + \frac{1}{Re} \nabla^2 V \]

where \( V = (u, v) \) is the velocity and \( p \) is the pressure. \( \text{Re} \) denotes the Reynolds number based on the velocity of the uniform flow, \( U_0 \), and the diameter. \( d \), of the circular cylinder. The reference scales for non-dimensionalization were \( d, U_0, \rho, \mu \) for the length, velocity, time, and pressure, respectively.

The numerical techniques adopted here are based on the well-known Marker and Cell (MAC) method, which was developed by [10]. The Poisson equation for the pressure, derived by taking the divergence of (2). All the spatial derivatives except those of the convection terms are approximated by second-order central differencing scheme. The non-linear terms are represented by means of a third order upwind scheme[11],

\[ \frac{\partial f_j}{\partial z} = U_0 f_{j+1} - 2f_j + f_{j-1} \]

where \( f \) is the pressure, and \( U_0 \) the pressure gradient. The Poisson equation for the pressure is solved iteratively by employing modified SOR method. For the marching Navier-Stokes equation, the semi-implicit scheme which is equivalent to the Euler backward scheme is used to integrate temporally.

In this paper, the following coordinate transformations, which includes the time variable are introduced

\[ \xi = \xi(x, y, t), \quad \eta = \eta(x, y, t), \quad \tau = \tau(x, y, t) \]

where \( x \) and \( y \) are the Cartesian coordinates and \( t \) is the time in the physical domain, \( \xi \) and \( \eta \) are the space co-ordinates and \( \tau \) the time in the computational domain.

The transformation of the coordinate derivatives are given by

\[ \begin{bmatrix} \frac{\partial_x}{\partial t} \\ \frac{\partial_y}{\partial t} \end{bmatrix} = \begin{bmatrix} \xi_x & \eta_x & \tau_x \\ \xi_y & \eta_y & \tau_y \end{bmatrix} \begin{bmatrix} \frac{\partial_x}{\partial t} \\ \frac{\partial_y}{\partial t} \end{bmatrix} \]

where subscripts denote the partial derivatives. The time variable is treated as

\[ \tau = t, \quad t_t = t_n = 0 \]

Substituting equation (6) into equation (5), one can obtain

\[ J \begin{bmatrix} \frac{\partial_x}{\partial t} \\ \frac{\partial_y}{\partial t} \end{bmatrix} = \begin{bmatrix} y_t & x_t & 0 \\ -x_t & y_t & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial_x}{\partial t} \\ \frac{\partial_y}{\partial t} \end{bmatrix} \]

where \( (x_t, y_t) \) denotes the velocity vector of a grid point and \( J \) is the Jacobian for the space coordinate transformation. The convective terms in the Navier-Stokes equations will be expressed as

\[ \frac{\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u}}{\partial_t} + \frac{1}{J} \left( \left( \mathbf{u} - \mathbf{v}_t, \mathbf{v} - \mathbf{v}_t \right) \mathbf{u} \right) \mathbf{u} 

The transformations given by the equation (8) implies that the moving boundaries in the physical domain become stationary in the computational domain.

RESULTS AND DISCUSSION

The entire computation were run on a Fujitsu FACOM VP200 Super Computer for the Reynolds number, \( \text{Re} = 1000 \). The results of the stationary circular cylinder are presented first to allow comparison with the experimental visualization. The results of the unsteady flow for a cylinder oscillates in the longitudinal direction are then presented. Figure 1(a) represents the development of the flow pattern computed for a stationary cylinder at time \( t = 100 \). The development of the vortex shedding is clearly seen. Two secondary vortices are formed, followed by a smaller single down stream of the separation point. As the vortices grow symmetrically, the shear layer joining the separation point to one of the vortices begins to develop instabilities and is drawn across the wake in response to the base pressure reduced by the action of the vortex growing across the wake. The stretching, diffusion, and dissipation of vorticity break up the deforming turbulent sheet and thereby the further supply of circulation to the vortices whose
rate of growth has already been reduced to its minimum. The vortex across the wake still continues to grow and entrains part of the oppositely signed vorticity. The shedding process for the second vortex does not commence until the critical wake during a given period undergoes successive interactions among themselves. This configuration agrees with the experimental and numerical results. The unsteady lift coefficient oscillates periodically. The periodic properties of the flow are clearly shown by the time-dependent evolution of the lift coefficient as shown in figure 1(b).

The numerical solutions of longitudinal oscillations of circular cylinder were performed. These results are for the case of the reduced velocity \( V_r = 3.0, 5.0, 7.2 \). Here, \( V_r \) is defined as \( V_r = U/f_d \), \( U = U_m \) where \( U_m \) is the maximum velocity of the oscillating cylinder. The amplitude of oscillation was set to be \( A(x/d) = 0.14, 0.17, 0.2, 0.26 \), where \( x \) refers to the displacement in \( x \)-direction. Figure 1(c) explains the flow visualization of the wake of oscillating cylinder reveals different patterns than the usual alternate Karman vortex street. In the lock-in range the influence of the flow oscillation is very strong. During one period of the flow oscillation, the vortex remains close to the cylinder as it forms and grows, even close in fact than before locking-on. The interesting feature of the phenomenon is the cylinder moves down stream from its mid position, the relative velocity of the cylinder increases from \( 0 \) to \( U \). When the cylinder reverses its direction, the relative velocity reaches and reaches a value of \( 2U \) at the mid-position of oscillation. The vortices move away from the cylinder under the influence of the uniform flow symmetrically at first and then becoming gradually asymmetrical. The resulting pattern is very complex. The streamwise interval of the vortices becomes large. Accordingly the value of lift is changing temporally and becomes complicated with including disturbances and different from the simple sinusoidal curve and are shown in figure 1(b) and 2. The mean inline force and its amplitude of oscillation were found to increase with increasing amplitude. The power spectra of the drag, lift and displacement is illustrated in figure 3. Synchronization can be seen when the driving frequency approaches in a range around double the Strouhal frequency.

REFERENCES

7. P. M. Rao, K. Kuwahara and K. Tsuboi, A direct simulation of the flow around a circular cylinder sinusoidally oscillating at low Kuehn-Carpenter numbers- Accepted 13th IMAC, Dublin.
Abstract: The paper deals with flow and heat transfer induced by a rotating circular disk when it is subjected to a mixed thermal boundary condition. By similarity analysis the governing equations are reduced to a system of ordinary differential equations. A transformation group has been found under which the reduced governing equations and the homogeneous boundary conditions are invariant. The thermal boundary condition is characterized by a nonnegative parameter \( m \); \( m = 0, 1, 0 \) correspond to the cases of prescribed temperature, prescribed heat flux and prescribed heat transfer coefficient (radiation boundary condition) respectively. If one wants to find solutions for different values of \( m \), there is no need to compute solution for each value of \( m \). If we know the solution for any particular value of \( m \), then this solution can be used to find the solution for any value of \( m \) by a simple method with the aid of the transformation group.

I. INTRODUCTION

The free convection and mixed convection on heated horizontal plates has received a great deal of attention [1-6]. These studies assume that the temperature at the plate is prescribed or the heat flux across the plate is prescribed. The study of convection problem when the plate is subjected to a mixed thermal boundary condition [7] has not received sufficient attention. The axisymmetric convection over a horizontal rotating disk subjected to mixed thermal boundary condition is governed by the boundary layer and \( \rho \) is the density respectively.

II. ANALYSIS

The convection over a rotating horizontal disk subjected to a mixed thermal boundary condition is governed by the boundary layer equations,

\[
\frac{\partial (ru)}{\partial T} + \frac{\partial (ru)}{\partial z} = 0
\]

\[
u \frac{\partial u}{\partial r} + \frac{ru}{r} - \frac{1}{\rho_m} \frac{\partial p}{\partial T} + \gamma \frac{\partial^2 u}{\partial z^2}
\]

\[
u \frac{\partial v}{\partial r} + \frac{ru}{r} + \frac{\partial w}{\partial z} - \gamma \frac{\partial^2 v}{\partial z^2} = 0
\]

\[
\frac{\partial p}{\partial z} + g \beta (T - T_w)
\]

with the boundary conditions,

\[
u = 0, v = rw, w = 0 \quad \text{at} \quad z = 0
\]

\[
u = 0, v = rw, p = p_m, T = T_w, \text{as} \quad z \to \infty
\]

\[
\alpha_s (T - T_w) = \alpha_T \frac{\partial T}{\partial z} = \alpha_p r^2 \quad \text{at} \quad z = 0
\]

where \( u, v \) and \( w \) are the velocity components in the radial, azimuthal and axial directions respectively. \( T \) is the temperature, \( p \) is the pressure, \( p_m, T_w, \rho_m \) are the ambient pressure, temperature and density respectively, \( \theta \) is the angular velocity of the rotating disk.

III. DISCUSSION AND CONCLUSION

The study of convection over a horizontal rotating disk subjected to a mixed thermal boundary condition has been reduced to solving the boundary value problem (10)-(16). The boundary condition (16) includes the following as special cases:

(1) Prescribed temperature (PT)

(2) Prescribed heat flux (PHF)

The symbols \( \rho, \beta, \nu \) and \( Pr \) denote the kinematic viscosity, gravitational acceleration, coefficient of thermal expansion, specific heat and Prandtl number respectively, \( a_s, a_T, a_p \) are prescribed constants.

Let us introduce the similarity transformation

\[
\eta = zL, \quad u = \frac{\sqrt{Fr}}{L} \quad q, \quad w = \frac{2 \sqrt{Fr}}{L} \quad F(\eta), \quad w = \frac{\sqrt{Fr}}{L} \quad G(\eta), \quad w = \frac{2 \sqrt{Fr}}{L} \quad F(\eta)
\]

\[
p - p_m - \frac{\rho r^2}{L^2} H(\eta), \quad T = \frac{\sqrt{Fr}}{L} \quad \theta(\eta)
\]

where \( A = g \sqrt{\gamma} \) and \( L \) is of diameter length is determined from the thermal boundary condition (8) in the manner as explained a little later. The equations (1) to (6) become

\[
F'' + 2FF' + F^2 + G^2 + 2H = 0
\]

\[
G'' + 2FG' + FG = 0
\]

\[
H' + \theta = 0
\]

\[
\theta'' + \frac{m^2 \theta'}{F} + \theta = 0
\]

\[
\theta(0) = \theta'(0) = 0
\]

where the primes denote differentiation with respect to \( \eta \), \( \Omega = \omega L \) is the rotation parameter (17)

\[
E = \frac{\alpha_s L}{\rho c}
\]

(18)

The solution of the boundary value problem (10)-(16) for any particular value of \( m \) can be obtained by shooting method, for any value of \( m \geq 0 \). It is interesting to note that the solutions corresponding to different values of \( m \) are dependent as stated in the following properties:

Property 1: The equations (10)-(15) are invariant under the transformation,

\[
\eta = A \eta, \quad \Omega = A^\Omega, \quad E = A^E, \quad F(\eta_1, \Omega_1, E_1) = F(\eta, \Omega, E) \]

\[
G(\eta_1, \Omega_1, E_1) = G(\eta, \Omega, E) \]

where \( A \) is any positive real number.

Property 2: If \( F(q, \Omega, E), G(q, \Omega, E), H(\eta, \Omega, E) \) and \( \theta(q, \Omega, E) \) is the solution of the boundary value problem (10)-(16) for any particular value of \( m \), say \( m=a \), then the solution for any value of \( m \) is given by eqns. (20) provided \( A \) is the positive root of the equation,

\[
A^m - (1-m) A + \theta(0, \Omega, E) = 0
\]

(21)

Property 3: If the solution of the boundary value problem (10)-(16) is same for any two distinct values of \( m \), then the solution is same for all values of \( m \).
Prescribed heat transfer coefficient (PHTC)

Hence \( \frac{\partial T}{\partial z} [\text{at } z=0] \) is replaced by

\[
N_u = -r \frac{\partial T}{\partial z} \bigg|_{z=0}
\]

The local Nusselt number \( N_u \) is defined by

\[
N_u = -r \frac{\theta'(0)}{\theta(0)}
\]

- \( N_u \) for \( PT \)
- \( N_u \) for \( PHF \)
- \( N_u \) for \( PHTC \)

The numerical results for different values of the parameters \( \Omega, E, m \) and \( \Pr=0.72 \), are given in Tables I and II. All the quantities of interest increase with \( m \). The temperature profiles are shown in Figures 1(a)-1(c). It is found that in the absence of dissipation, increase in rotation \( \Omega \) decreases the temperature, whereas increase in dissipation \( E \) increases the temperature near the disk and it is more pronounced with \( \Omega \). Increase in \( m \) increases the temperature largely and decreases the thermal boundary layer thickness. Table III gives the values of \( \Delta \) needed for transition from one case to the other with the aid of equation (20). The following example illustrates the transition.

For \( PT \) with \( \Omega=1, E = 0.1 \)

\( \theta(0) = -0.7451, F'(0) = 1.3772 \)

which gives

\( A = 0.9521, \Omega = 1.1031, E = 0.1050 \)

\( \theta(0) = 1.2779, F'(0) = 1.5897 \)

\( \theta(0) = -1.0735, F'(0) = 1.1848 \)

\( A = 0.7451, \Omega = 1.8012, E = 0.1342 \)

\( \theta(0) = 4.3544, F'(0) = 3.3172 \)

\( \theta(0) = -2.2444, F'(0) = 3.1595 \)

It is interesting to know that the solution is independent of \( m \) for certain values of \( \Omega \), \( E \) and \( \Pr \). Table IV shows the values of \( \Pr \) for \( E=0 \), for which the solution is independent of \( m \).

ACKNOWLEDGEMENT: One of the authors (V. Kumaran) is thankful to Council of Scientific and Industrial Research, New Delhi, India for the financial assistance.

Table I

<table>
<thead>
<tr>
<th>( \Omega )</th>
<th>( E )</th>
<th>( m )</th>
<th>( \theta(0) )</th>
<th>( \theta'(0) )</th>
<th>( \theta''(0)/\theta(0) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0</td>
<td>1.0000</td>
<td>0.7570</td>
<td>0.7570</td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>0.5</td>
<td>1.1260</td>
<td>0.8734</td>
<td>0.7573</td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>1</td>
<td>1.2610</td>
<td>1.0000</td>
<td>0.7930</td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>( \infty )</td>
<td>4.0214</td>
<td>4.0214</td>
<td>1.0000</td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>0</td>
<td>1.0000</td>
<td>0.7410</td>
<td>0.7410</td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>0.5</td>
<td>1.1530</td>
<td>0.8640</td>
<td>0.7505</td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>1</td>
<td>1.2827</td>
<td>1.0000</td>
<td>0.7796</td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>( \infty )</td>
<td>4.3543</td>
<td>4.3543</td>
<td>1.0000</td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>0</td>
<td>1.0000</td>
<td>0.7575</td>
<td>0.7575</td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>0.5</td>
<td>1.1260</td>
<td>0.8737</td>
<td>0.7577</td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>1</td>
<td>1.2610</td>
<td>1.0000</td>
<td>0.7933</td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>( \infty )</td>
<td>4.0179</td>
<td>4.0179</td>
<td>1.0000</td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>0</td>
<td>1.0000</td>
<td>0.7411</td>
<td>0.7411</td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>0.5</td>
<td>1.1530</td>
<td>0.8640</td>
<td>0.7506</td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>1</td>
<td>1.2827</td>
<td>1.0000</td>
<td>0.7797</td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>( \infty )</td>
<td>4.3527</td>
<td>4.3527</td>
<td>1.0000</td>
<td></td>
</tr>
</tbody>
</table>

Table II

<table>
<thead>
<tr>
<th>( \Omega )</th>
<th>( E )</th>
<th>( m )</th>
<th>( F'(0) )</th>
<th>( -G'(0) )</th>
<th>( H'(0) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0</td>
<td>1.0966</td>
<td>0</td>
<td>0.9933</td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>0.5</td>
<td>1.1778</td>
<td>0</td>
<td>1.1093</td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>( \infty )</td>
<td>2.5273</td>
<td>0</td>
<td>3.0241</td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>0</td>
<td>1.1023</td>
<td>0</td>
<td>0.9976</td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>0.5</td>
<td>1.1803</td>
<td>0</td>
<td>1.1046</td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>( \infty )</td>
<td>2.6611</td>
<td>0</td>
<td>3.2329</td>
<td></td>
</tr>
</tbody>
</table>

Table III

<table>
<thead>
<tr>
<th>( \Omega )</th>
<th>( F'(0) )</th>
<th>( -G'(0) )</th>
<th>( H'(0) )</th>
<th>( \Pr )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.6270</td>
<td>0.0000</td>
<td>0.7100</td>
<td>2.4372</td>
</tr>
<tr>
<td>0.1</td>
<td>0.6616</td>
<td>0.0669</td>
<td>0.7101</td>
<td>2.4215</td>
</tr>
<tr>
<td>0.5</td>
<td>0.7553</td>
<td>0.3552</td>
<td>0.7313</td>
<td>2.1213</td>
</tr>
<tr>
<td>1.0</td>
<td>1.0481</td>
<td>0.8004</td>
<td>0.7252</td>
<td>1.0273</td>
</tr>
</tbody>
</table>

Fig. 1 The temperature profiles for (a) \( PT \), (b) \( PHF \), (c) \( PHTC \) for different values of \( \Omega \) and \( E \).
Abstract Self adaptive numerical solution methods for combustion problems are developed and results are presented for physically idealized situations. First, the treatment of the time dependence is discussed separately, as is the development and implementation of spatial adaptive gridding techniques, representative numerical results based these two major ingredients of any numerical technique for the simulation of ignition problems are presented. Secondly, time-dependence and spatially two-dimensional variations are treated simultaneously; as a representative example self-ignition in a premixed reactive mixture is numerically simulated.

1. INTRODUCTION

In combustion and related areas numerical techniques are needed which are able to produce high spatial and spatiotemporal resolution of flame structures and evolutions; only such ability will provide enhanced insight into the many modes and ingredients of combustion processes. The use of adaptive methods, i.e., of methods that "automatically" adapt the computational mesh in some "optimal" way to the specific problem under consideration, is the desirable numerical approach to all engineering problems; it is a must, however, for the computation of reactive flows. The physical reason for this is the heat-release associated with any combustion process. As a consequence of the heat release, profiles of quantities involved in combustion processes typically exhibit steep gradients and strong curvature, thus necessitating the use of adaptive methods in order to control both the temporal and spatial discretization errors. For time independent steady problems in combustion, adaptive methods have proven useful and are being used, see e.g. /1 3/. Apart from notable exceptions /4,5/, little attention has been given to time dependent, two-dimensional combustion problems. Therefore, in the present paper we outline possible approaches to fully implicit adaptive algorithms suitable for successfully tackling the numerical simulation of unsteady, spatially two-dimensional reactive flows.

Firstly, we consider briefly techniques suitable for tackling the time-dependence of combustion phenomena and present two spatially one-dimensional examples, viz., a pulsating flame and a self ignition process taking place in a non-premixed flow. Secondly, we consider spatially two-dimensional, steady problems focusing attention on the development of self adaptive meshing procedures. As a representative example, numerical results are presented for a laminar flame propagating in a strained mixing layer. Thirdly, we combine the seperately developed ingredients for the simulation of time-dependent and spatial effects, and consider truly time-dependent combustion in two space dimensions. For the latter case, hot-spot-like self-ignition taking place in an originally cold, reactive mixture provides the physical basis for a numerical example.

2. FORMULATION

2.1 Differential equations

The general class of differential equations to be considered herein can be written in the form

\[ u_t = f(x,y,t,u,u_x,u_y,u_{xx},u_{yy}) \quad \text{for} \ (x,y) \in \Omega, \ t > 0, \]

\[ r(x,y,t,u(x,y,t),u_t(x,y,t),u_{x}(x,y,t),u_{y}(x,y,t)) = 0 \]

\[ \text{for} \ (x,y) \in \Omega \cup \partial \Omega, \ t > 0, \]

\[ u(x,y,0) = v(x,y) \quad \text{for} \ (x,y) \in \Omega \cup \partial \Omega, \]

where \( u, f, r \) and \( v \) are \( N \)-vectors, \( x \) and \( y \) are the cartesian space coordinates and \( t \) the time; \( \Omega \) denotes a rectangular domain of integration and \( \partial \Omega \) its boundary. Equation (1) represents a nonlinear parabolic mixed initial boundary value problem for the \( N \) dependent variables represented by \( u \).

2.2 Difference equations

The integration of system (1) with respect to time is performed in steps starting with specified profiles [ which, in general, should satisfy the governing equations ] at time level \( n = 0 \) with \( t = t^0 = 0 \). Solutions to (1) are sought at the subsequent time levels \( (n = 1, t = t^1), (n = 2, t = t^2), \) and so on, with \( 0 = t^0 < t^1 < t^2 < \cdots < t^n < \cdots \), where here and below the superscript \( n \) is used to identify quantities at time level \( n, n = 0,1,2, \ldots \). The integration of (1) is considered complete if either a specified level \( n_{max} \) or a specified time \( t_{max} \) is reached. With respect to the space variables \( x \) and \( y \), system (1) is discretized on a mesh \( M^n \) of grid points,

\[ M^n = \{(x_{1}^n,y_{1}^n), (x_{2}^n,y_{2}^n) \cdots (x_{m}^n,y_{m}^n)\}. \]

Note that the mesh may or may not be a simple tensor product grid. In fact, for the simulation of combustion problems it regular grids are desirable which locally and instantaneously concentrate grid points where they are needed, namely in regions or spots of high spatial and temporal activity. A suit-
able adaptive-meshing strategy is outlined below. If the spatial domain of integration is infinitely large with respect to one or both coordinate directions, for example extending from \( z = -\infty \) to \( z = +\infty \), then \( a_0 \) and \( a_0 \) are chosen as positive and sufficiently large to ensure that the conditions at either boundary can asymptotically be satisfied /6/, an infinite domain of integration with respect to \( y \) is treated similarly.

3. NUMERICAL METHODS

Subsequently the symbols \( U \) and \( F \) will be used to denote the \( m \times N \)-vectors which result from the spatial discretization of the \( N \)-vectors \( u \) and \( f \), respectively. For the solution of time-dependent problems at each time step or steady-state problems (not considered herein), Newton's method can be applied to the system of nonlinear equations, \( F(U) \), which results from the discretization of the governing equations. Note that both \( U \) and \( F \) depend on the particular time level \( n \) under consideration. Thus, the linear system

\[
J(U^k)(U^{k+1} - U^k) = -\omega_k F(U^k), \quad k = 0,1,\ldots ,
\]

(4)
is solved where \( U^k \) denotes the solution after \( k \) Newton iterations, and \( \omega_k \) and \( J(U^k) \) are the damping parameter and the Jacobian matrix, respectively, based on \( U^k \). The damping strategy /7,8/ allows the Jacobian, which is generated numerically, to be re-evaluated only periodically /8/.

Upon discretising the governing equations only with respect to space, one is left with a large system of DAEs (differential-algebraic equations). In recent years various algorithms have been developed for the solution of such systems as well as complete software packages. Examples for the latter are \textit{DASSL} /9/ and \textit{LIMEX} /10,11/ developed at the Sandia National Labs., USA, and at the University of Heidelberg, West Germany, respectively. \textit{DASSL} uses backward-differentiation formulas, \textit{LIMEX} a newly developed extrapolation method particularly designed to deal with stiff systems. Since, in a sense, software packages are multi-purpose codes, they are not optimized with regard to a specific problem under consideration. Therefore, we have developed a new code particularly designed for the solution of systems of DAEs that arise in one-dimensional combustion problems which, however, uses some of the basic elements of \textit{LIMEX}.

Regardless whether Newton's method or the method of lines is employed, the solution of system (1) at time level \( n \) depends on the solution at level \( n-1 \) taken at the grid points of mesh \( M^n \). Since in general the grids at levels \( n-1 \) and \( n \) are not the same, the solution obtained at level \( n-1 \) on grid \( M^{n-1} \) must be interpolated onto grid \( M^n \) which, regardless of the interpolation procedure, introduces an additional spatial discretization error into the algorithm. We use piecewise monotonic cubic Hermite interpolation /12/.

4. ADAPTIVE SELECTION OF GRID POINTS

The procedures and criteria for the adaptive selection of grid points are of critical importance to the efficiency of the algorithms that are used for the solution of combustion problems. In particular, strategies are required that place the grid points where they are needed in order to bound the local space discretization error. In generalization of procedures outlined previously for the adaptive computation of steady one-dimensional combustion problems, for any fixed time level \( n \) we equidistribute the mesh \( M^n \) on intervals \( h^*_1, h^*_2 \) with respect to a non-negative weight function \( W^n \) and a constant \( C^n \). Here the intervals in the \( x \) and \( y \) direction, \( h^*_1, h^*_2 \), are of lengths characteristic of the local and instantaneous spacing of mesh points in the vicinity of point \( (x^n, y^n) \). For instance, for the \( x \) direction \( W^n \) is selected such that

\[
\int_{h^*_1}^{h^*_2} W^n dx = C^n, \tag{5}
\]

where \( m = m^n \). Specifically, the weight function \( W^n \) is chosen as

\[
W^n = \max_{1 \leq k \leq 2N+1} W^n_k, \tag{6a}
\]

where

\[
W^n_k = \frac{[\partial U_k/\partial x]}{g^n[\max U_k - \min U_k]}, \quad 1 \leq k \leq N \tag{6b}
\]

and

\[
W^n_{2N+1} = d^n. \tag{6d}
\]

In Eqs. (6b) and (6c) "\( \min \)" and "\( \max \)" stand for the minimum and maximum value in the interval \( a_s \leq x \leq a_e \) of the respective quantity, and \( g^n \) and \( c^n \) are positive scaling factors; their numerical values are less than unity if in Eq. (5) \( C^n = 1 \) is employed. In Eq. (6d), \( d^n \) is a positive constant which represents the maximum size of any interval \( h_i \). To prevent the size of adjacent mesh intervals from varying too much, we require that at any time level \( n \) the mesh be locally bounded, i.e.,

\[
R^{-1} \leq h_j / h_{j-1} \leq R, \tag{7}
\]

where \( R \) is a constant greater than one. The equidistribution procedure with respect to the \( y \) direction is performed analogously. The adaptive griding procedure to be carried out at each time level \( n \) essentially consists of \( T \) steps which, because space is restricted in the present volume, can be presented only in the full-length version of the paper.

5. EXAMPLES, RESULTS AND DISCUSSION

As a first example we consider a problem, the so-called test problem A, proposed for a GAMM Workshop at Technical University Aachen, West Germany, viz., an unsteadily propagating flame with one step chemistry and Lewis number different from unity /13/. The governing equations are

\[
\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} + \eta, \tag{8}
\]

\[
\frac{\partial Y}{\partial t} = \frac{1}{Le} \frac{\partial^2 Y}{\partial x^2} - R, \tag{9}
\]

where \( T \) and \( Y \) denote a normalized temperature and mass fraction, respectively, and where

899
Time \( t \) and space-variable \( x \) are suitably nondimensionalized. In the rate expression (10), \( \alpha \) and \( \beta \) denote a nondimensional heat-release parameter and a nondimensional activation energy, respectively. The initial conditions are given by

\[
T = \exp(x), \quad x \leq 0, \quad (11.a)
\]
\[
Y = 1 - \exp(Le \ x), \quad x \leq 0, \quad (11.b)
\]
and

\[
T = 1, \quad x > 0, \quad (11.c)
\]
\[
Y = 0, \quad x > 0, \quad (11.d)
\]
the boundary conditions by

\[
T = Y = 1 = 0 \text{ as } x \to -\infty, \quad (11.e)
\]
\[
\partial T/\partial x = \partial Y/\partial x = 0 \text{ as } x \to +\infty. \quad (11.f)
\]

In all calculations \( \alpha = 0.8 \) is adopted; \( \beta \) and \( Le \) are taken as variable parameters.

As one example, shown in Fig. 1 is the computed propagation velocity \( v_F \) of the flame as a function of time for \( Le = 1.45 \) and \( \beta = 32 \). It is seen that after a short initial period a continuous limiting cycle results. To accurately resolve the spatial structure of this pulsating flame the adaptive meshing procedure outlined above requires 80 to 100 grid-points; to obtain comparable resolution with an equidistant grid as many as 800 to 1000 grid points would be required.

As a second example we consider auto-ignition in a non-premixed flow generated by directing a hot air stream (\( T = 800K \)) and a cold fuel stream (\( T = 300K \)) towards each other.

[Image: Oscillating flame velocity \( v_F \) as a function of time \( t \).]

The governing equations are the conservation equations of overall mass, species mass, momentum and energy, space limitations do not allow to present these equations here. Chemistry is assumed to occur via the overall one-step reaction \( F + vO_2 \rightarrow P \). For a detailed discussion of appropriate initial and boundary conditions, and the physical background ref. [14] should be consulted.

As a representative result, Fig. 2 shows temperature profiles during the ignition process. It is seen that a continuous transition takes place from inert mixing at initial time \( t = 0 \) to a steadily burning diffusion flame. The steadily burning state is reached after roughly 10 microseconds.

[Image: Temperature profiles during a truly time-dependent auto-ignition process. Data: \( \alpha = 53.7 \text{ s}^{-1}, \quad p = 40 \text{ bars}, \quad \text{steady-state ignition point at } \alpha = 58.7 \text{s}^{-1} \). The frozen initial profiles pertain to \( t = 0 \), the other profiles to \( t = 1.01 \times 10^{-9} \text{s}, \quad 3.82 \times 10^{-9} \text{s}, \quad 6.87 \times 10^{-9} \text{s}, \quad 1.57 \times 10^{-8} \text{s}, \quad 3.43 \times 10^{-8} \text{s}, \quad 7.66 \times 10^{-8} \text{s}, \quad 1.71 \times 10^{-7} \text{s}, \quad 3.86 \times 10^{-7} \text{s}, \quad 3.01 \times 10^{-7} \text{s}, \quad 2.25 \times 10^{-6} \text{s}, \quad 6.25 \times 10^{-6} \text{s}. \)

As a third example we consider a flame propagating in a strained mixing layer. Such flames can be generated, for instance, in a Tsuji like counterflow geometry. The governing equations for this problem were derived by Liénard [12], viz.,

\[
u_F \frac{\partial Y_F}{\partial x} - Y_F \frac{\partial v_F}{\partial y} = \frac{\partial^2 Y_F}{\partial x^2} + \frac{\partial^2 Y_F}{\partial y^2} - \delta \beta^3 Y_F Y_O e^{\delta(T-T_M)/(T_F-T_o)} \quad (11)
\]

\[
u_F \frac{\partial Z}{\partial x} - Y_F \frac{\partial Z}{\partial y} = \frac{\partial^2 Z}{\partial x^2} + \frac{\partial^2 Z}{\partial y^2},
\]

\[
T - T_o = 1 - Y_F - Y_O,
\]

\[
Z = \frac{s Y_F + 1 - Y_O}{s + 1},
\]

with the boundary conditions

\[
y \to -\infty : Y_F = Z = 0,
\]
\[
y \to +\infty : Y_F = Z = 1,
\]
\[
x \to -\infty : Z = \frac{1}{2} \left( s + 1 \right) Y_F + 1 - \frac{1}{s + 1} Y_O e^{\delta(y/\sqrt{2})}
\]
\[
x \to +\infty : \text{zero } x \text{ gradients for all dependent variables.}
\]

The unknowns in this problem are the (constant) burning-rate eigenvalue \( u_F \), the temperature \( T \), and the mass fractions of fuel and oxidiser, \( Y_F \) and \( Y_O \), respectively. The Damköhler number \( \delta \) and the nondimensional activation energy \( \beta \) are variable parameters to be specified, for the results presented herein we have selected \( \delta = 0.5 \) and \( \beta = 5 \).
Shown in Fig. 3 is the computational mesh on which the final solution has been obtained for the above problem. It is

![Figure 3: Adaptively generated, converged computational mesh for Example 3.](image)

seen that the mesh has a tree-like structure due to the generation of individual mesh elements at successive levels \( z \) of the adaptive meshing procedure. As one example, shown in Fig. 4 is a surface plot of temperature. Notice that mesh points are concentrated in regions where the temperature exhibits steep gradients, and where the surfaces of both temperature and fuel mass fraction have strong curvature.

![Figure 4: Surface plot of temperature for Example 3.](image)

6. CONCLUSIONS

We have developed and discussed numerical approaches having all the ingredients necessary for successfully tackling problems typically arising in combustion. Specifically, emphasis has been laid on self-adaptive gridding procedures applicable to time-dependent two-dimensional reactive flows. As examples pulsating flame propagation, auto-ignition in a non-premixed flow, flame propagation in a strained mixing layer and hot-spot-like self-ignition have been considered.

REFERENCES

A DOMAIN-ADAPTIVE BLOCK BIDIAGONAL TECHNIQUE FOR ANNULAR LIQUID JETS

J. I. Ramos
Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213-3890
U.S.A.

Juan Folgueras and Enrique Nava
F. Informática / E.T.S.I. Telecomunicación
Universidad de Málaga
Pitso El Ejido, s/n
29013-Málaga
SPAIN

ABSTRACT - An adaptive, block-bidiagonal finite difference method is used to study the response of annular liquid jets to the injection of mass into the volume enclosed by the annular jet. It is shown that the annular jet's response is characterized by damped oscillations in both the convergence length and the pressure of the gases enclosed by the jet, and that the amplitude and number of these oscillations increase as the initial pressure ratio across the annular jet and the pressure of the gases surrounding the jet are increased.

I INTRODUCTION
Annular liquid jets may form enclosed volumes if the pressure difference between the gases enclosed by and surrounding them is overcome by surface tension, and can be used to measure the dynamic surface tension of liquids and burn toxic wastes in the volume enclosed by the annular jet [1].

The equations which govern the fluid dynamics of inviscid, isothermal, annular liquid jets were derived in Reference [2]. These equations are asymptotic to terms proportional to the annular jet's thickness-to-radius ratio at the nozzle exit and are valid for steady and unsteady jets. In this paper, an adaptive block-bidiagonal finite difference technique is used to study the dynamic response of annular liquid jets to the injection of mass into the volume enclosed by the annular jet as a function of the initial pressure difference between the gases enclosed by and surrounding the jet, and pressure of the gases surrounding the annular jet.

II FLUID DYNAMICS EQUATIONS
The nondimensional equations governing the fluid dynamics of inviscid, isothermal, annular liquid jets can be written as [1]

\[ \frac{\partial U}{\partial t} + \frac{\partial F}{\partial U} \frac{\partial U}{\partial z} = G \]

where

\[ U = [m, mR, mU, m\bar{U}]^T, \quad F = [mU, mRmU, mUm\bar{U}]^T \]

\[ G = \left[ 0, \frac{m}{R}, \frac{1}{W_e} \left( \frac{\partial F}{\partial U} - C_{pm} \frac{\partial R}{\partial z} \right), \frac{1}{W_e} \left( C_{pm} \frac{\partial R}{\partial z} - \frac{\partial F}{\partial U} \right) \right]^T \]

\[ U(\tau, 0) = [1, 1, 1, \tan \theta_0]^T \]

\[ \tau = \frac{u_0^2}{g R_0}, \quad W_e = \frac{m_0 u_0^2}{2 \sigma R_0} \]

\[ C_{pm} = C_p W_e, \quad C_p = \left( \frac{p_i}{p_0} \right) \frac{R_0^2}{m_0 u_0^2} \]

\[ J = R \left[ 1 + \left( \frac{\partial R}{\partial z} \right)^2 \right]^\frac{1}{2} \]

If the gases enclosed by the annular liquid jet are ideal and isothermal, the liquid does not absorb the gases that it encloses, and mass is injected into the volume enclosed by the jet at a rate \( a \) and during a time \( t_{inj} \), then

\[ \frac{p_i(t)}{p_0} = m_i(t) \left( \frac{V(t)}{V(0)} \right) \left( \frac{p_i(0)}{p_0} \right) \]

\[ m_i(t) = m_i(t_{inj}) = 1 + a t, \quad 0 \leq t \leq t_{inj} \]

\[ m_i(t) = 1 + a t_{inj}, \quad t > t_{inj} \]

Substitution of Eq. (8) into Eq. (6) yields

\[ C_{pm} = C_{pm} \left[ m_i(t) \left( \frac{V(t)}{V(0)} \right) \left( \frac{p_i(0)}{p_0} \right) - 1 \right] \]

where

\[ C_{pm} = \frac{p_i R_0^2}{2 \sigma}, \quad V(t) = \int_0^L \frac{R(t)}{R(t)} dz \]

\[ V = \frac{V_0}{R_0^2}, \quad L(t) \text{ is the axial distance at which the annular jet's inner interface radius is zero, i.e.,} \]

\[ P(t, L(t)) = 0 \]

and

\[ b = \frac{m b_0}{R R_0}, \quad R_a = R - b/2 \]

III DOMAIN-ADAPTIVE TECHNIQUE
The annular liquid jet geometry is curvilinear and time-dependent and has an unknown, time-dependent, downstream boundary, i.e., the convergence point. This geometry can be transformed into a unit interval by means of the mapping

\[ (t, \eta) \rightarrow (\tau, \eta), \quad \tau = t, \quad \eta = z/L \]

The Jacobian of this mapping is \( L(t) \) which is a function of time. Substitution of Eq. (15) into Eq. (1) yields

\[ \frac{\partial U}{\partial \tau} + H \frac{\partial U}{\partial \eta} = G, \quad H = \frac{1}{L} \left[ \frac{\partial F}{\partial U} - \eta \frac{dL}{dt} \right] \]

where \( I \) is the unit matrix. Equation (16) can be discretized in an equally spaced grid such that \( \eta_1 = 0 \) and \( \eta_f = 1 \) where \( f \) denotes
the number of grid points, by means of backward differences for the advection term, and central differences for $G$, and the resulting $O(\Delta r, \Delta\eta)$-accurate finite difference equation can be written as

$$-C_i^j U_i^{j+1} + (I + C_i^j) U_i^j + U_i^j = \Delta r G_i^{j+1}$$

(17)

where $i = 2, 3, \ldots, I - 1$, and

$$C = H \Delta r / \Delta\eta$$

(18)

Since both $H$ and $C$ depend on $dL/d\tau$ and $L$, an equation must be obtained for the convergence length. Such an equation can be obtained as follows. At the convergence point (cf. Eqs. (13) and (14))

$$R_i(t, z = L(t)) = 0, \quad b_i(t, z = L(t)) = 2R_i(t, z = L(t))$$

(19)

Therefore, the following algebraic equation must be satisfied at the convergence point (cf. Eq.(14))

$$2R_i^2(t, z = L(t)) = b_0 m(t, z = L(t))/R$$

(20)

Differentiation of Eq. (20) with respect to $t$, and use of Eqs. (1) and (15) yield

$$\frac{dL}{dt} = \frac{4R_i^2(\frac{\partial R}{\partial \eta} - \eta L) - b_0 \frac{\partial}{\partial \eta} (mu)}{4R_i^2 \frac{\partial R}{\partial \eta}}$$

at $\eta = 1$

(21)

which is an ordinary differential equation for $L$ and which can be discretized, in an equally-spaced grid, as

$$\frac{dL}{dt} = \frac{4R_i(\frac{R_i - R_{i-1}}{\Delta\eta} - \eta_i L) - b_0 (\frac{mu}{R}) \frac{R_i}{\Delta\eta}}{4R_i \frac{R_i - R_{i-1}}{\Delta\eta} - b_0 \frac{m_i - m_{i-1}}{\Delta\eta}}$$

(22)

The values of $u, R, V$ and $m$ at $i = I$ can be calculated by linear extrapolation as

$$U_I = 2U_{I-1} - U_{I-2}$$

(23)

**IV PRESENTATION OF RESULTS**

Figures 2 and 3 illustrate the effects of the initial pressure ratio across the annular liquid membrane on both the convergence length and the pressure coefficient. These figures correspond to nonpressurized and overpressurized annular membranes, respectively, and indicate that the initial pressure ratio across the membrane has a great effect on its dynamic response. In particular, Figures 2 and 3 clearly indicate that the time required to reach asymptotic, steady state after the end of mass loading increases as the initial pressure ratio across the membrane is increased, and that overpressurized membranes exhibit damped oscillations analogous to those of a mass-spring-dashpot system.

Figures 2 and 4 illustrate the effects of $C_{pm}$ on the response of annular membranes subject to mass loading, and indicates that the maximum values of both the pressure coefficient and the convergence length, and the time required to reach asymptotic, steady state after the end of mass injection increase as $C_{pm}$ is increased. Figure 4 also shows that the critical pressure coefficient of unity determined from the solution of the steady state governing equations [2] can be exceeded without affecting the stability of the annular membrane, and that the initial response of the pressure coefficient is nearly linear.

**V CONCLUSIONS**

The dynamic response of annular liquid jets to mass loading has been analyzed by means of an adaptive finite difference method that transforms the unknown, time-dependent, curvilinear geometry of the annular jet into a unit interval, and that yields a differential equation for the convergence length, i.e., for the axial distance at which the annular jet merges on the symmetry axis to form a solid jet. A block-bidiagonal technique has been used to determine the annular jet mean radius, mass per unit length, and axial and radial velocity components of the liquid in an iterative manner.

It has been shown that the pressure coefficient and the pressure of the gases enclosed by the annular liquid jet respond instantaneously to the mass injection, whereas there is a lag in the response of the convergence length. This lag is due to the inertia of the jet and the assumption that the gases enclosed by the jet are isothermal, and decreases as the injection duration is increased.

**REFERENCES**


Diffusion Current in MM-Wave DDRs

S.P. Pati and G.N. Dash

Department of Physics, Sambalpur University, Jyoti Vihar, Sambalpur-768019 (Orissa) INDIA

Abstract:
Diffusion of charge carriers would become a prominent physical phenomenon in case of short mm-wave DDRs due to narrow depletion zone and high carrier concentration gradient. An economical numerical simulation method to compute the diffusion current and the magnitude of diode negative resistance of Si DDRs due to diffusion current density is presented which provides a realistic picture of the effect of carrier diffusion in IMPATT devices for mm-wave operation.

Introduction:
Ultrathin depletion layer and high mobile space charge concentration in IMPATT diodes operating in mm-wave and short mm-wave regions would push the carrier diffusion current to comparable limit of drift current and both drift and diffusion currents would contribute to microwave negative resistance generated in the device. The inclusion of diffusion current to the mathematical analysis of the device physics is often neglected due to complexities that may be involved in the analysis. The authors have devised a comparative, low-cost numerical simulation method to solve IMPATT device equations under high frequency small signal conditions by considering both drift and diffusion currents. Our analysis also can give the microwave at the p-side boundaries of the diode. These are given by

\( \mathcal{E}(\mathcal{R} - \mathcal{O}) - \mathcal{O} = 0 \)

and

\( \mathcal{O}X + \mathcal{O}P = 0 \)

at the p-side boundary.

Assuming that the diffusion current is a small perturbation on the drift component, Equ (6) can be written as

\( (L + \lambda L') Z = \mathcal{F} \)

Where \( L' \) contains all the terms involving \( \mathcal{D}_p \mathcal{N}_p \) and \( \mathcal{D}_p \mathcal{N}_p \) only on the L.H.S. of Equ (6) and \( \mathcal{F} \) is the R.H.S. of Equ (6).

After separating into real and imaginary parts, Equ (10) is first solved and then Equ (11) are solved progressively for \( i = 1, 2, 3, \ldots \) one can write Equ (9) as

\( (L + \lambda L') (Z_0 + \lambda Z_1 + \lambda^2 Z_2 + \ldots \ldots ) = \mathcal{F} \)

Equating coefficients of various powers of \( \lambda \) a series of differential equation is obtained as

\( L^2 Z_0 = \mathcal{F} \)

and

\( L^2 Z_i = - L Z_{i-1}, \quad i = 1, 2, 3 \ldots \ldots \)

After separating into real and imaginary parts, Equ (10) is first solved and then Equ (11) are solved progressively for \( i = 1, 2, 3 \ldots \ldots \) etc. using the numerical technique described as follows.

Computer Method:
Equ (10) is solved by a modified Runge-Kutta algorithm following an iterative procedure. The iterations over the initial values of resistance \( R_0 \) and \( X_0 \) may be varied in the four possible ways i.e., \( R_0 R_2 \Delta R_0 \) and \( X_0 X_2 \Delta X_0 \). The logic in which the initial values of \( R_0 \) and \( X_0 \) are to be varied to obtain the required boundary conditions at the other edge depends on the structure of the diode as well as on the frequency of operation. The programme software has been designed in such a way that the variations in the initial values of \( R_0 \) and \( X_0 \) automatically switch over to the converging track very swiftly. The software is thus free from numerical instability. The accuracy limit is set at
Fig. 1 Microwave diode resistance contributed by diffusion current in Silicon DDRs.

0.02%. After the solution of Eqn (10) the quantities on R.H.S. of the Eqn (11) for i=1 are obtained from the knowledge of \( R_0 \) and \( \chi_0 \) at each space point. The third and fourth order derivatives of \( R_0 \) and \( \chi_0 \) are obtained numerically following Sterling's formula. Then Eqn (11) is solved for i=1 following the same Runge-Kutta approach to get the first order diffusion correction. The different order of diffusion corrections may be obtained by progressively solving Eqn (11) for i=2, 3, 4, etc. which gives the diffusion contribution to the negative resistance of the diode.

RESULTS:

Flat profile Si DDRs for operation in V, F, D, G and Y bands with centre frequencies respectively at 60 94, 140, 170 and 220 GHz are designed following a static analysis (2). The small signal mm-wave properties like diode negative conductance (G) and diode negative resistance \( Z_R \) which are determined following our method are presented in Table-1 for the cases (i) when diffusion is neglected and (ii) when diffusion is considered. It is seen that the effect of carrier diffusion on the device characteristics remains marginal for diodes with frequency of operation below 100 GHz.

Further it is observed that diffusion current enhances the device negative resistance for diodes designed to operate below 150 GHz. The degrading effect of carrier diffusion starts with diodes designed to operate above 150 GHz and it substantially reduces the device negative resistance for Y-band operation. The device negative conductance also records a progressive deterioration due to carrier diffusion as one goes from low frequency V-band to high frequency Y-band. The distribution of resistance, contributed by diffusion current, in the depletion layers of different diodes are shown in Fig. 1. The curves show that the diffusion contribution to diode resistance becomes positive for G and Y band diodes giving rise to decrease in the device negative resistance for high frequency mm-wave bands leading to fall in the power output and efficiency of IMPATT devices for short mm-wave operation.

**Table-1**

<table>
<thead>
<tr>
<th>Band Optimum Frequency</th>
<th>( D_n ), ( n_0 )</th>
<th>( Z_R )</th>
<th>( D_n ), ( n_0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>GHz</td>
<td>( \times 10^{-6} )</td>
<td>( \times 10^{-9} )</td>
<td>( \times 10^{-6} )</td>
</tr>
<tr>
<td>V</td>
<td>60</td>
<td>7.8</td>
<td>7.6</td>
</tr>
<tr>
<td>F</td>
<td>105</td>
<td>38.1</td>
<td>6.4</td>
</tr>
<tr>
<td>D</td>
<td>130</td>
<td>56.8</td>
<td>2.3</td>
</tr>
<tr>
<td>G</td>
<td>170</td>
<td>99.1</td>
<td>1.4</td>
</tr>
<tr>
<td>Y</td>
<td>220</td>
<td>192</td>
<td>1.0</td>
</tr>
</tbody>
</table>

REFERENCES:

Abstract The equations governing linear wave propagation in viscoelastic media, either single-phase or multiphase, can be written as a single first-order matricial differential equation in time. The formal solution is the evolution operator $e^{\mathbf{H} t}$ acting on the initial condition vector, where $\mathbf{H}$ is a linear operator matrix containing the spatial derivatives and medium properties, and $t$ is the time variable. The problem is solved numerically approximating the evolution operator by an optimal polynomial expansion depending on the location of the eigenvalues of $\mathbf{H}$ in the complex frequency plane. The eigenvalue analysis is carried out for the anisotropic-viscoelastic and porous viscoacoustic constitutive relations and respective limiting rheologies. For each case an optimal expansion of the evolution operator is identified, which provides highly accurate solutions and fast convergence compared to Taylor expansion or temporal differencing.

I. INTRODUCTION

Linear viscoelasticity provides a general framework for describing the anelastic effects in wave propagation, i.e., the conversion of part of the energy into heat, and the dispersion of the wave field Fourier components with increasing time. A dissipation model which is consistent with real materials is the general standard linear solid which is based on a spectrum of relaxation mechanisms. However, implementation of this rheology in the time domain is not straightforward due to the presence of convolutional kernels (Holzmann's superposition principle). To avoid the time convolutions, it is necessary to introduce into the formulation additional variables, called memory variables in virtue of their nature [1]-[5]. The wave equation of the medium can be written as a first-order differential equation in time as

$$\dot{\mathbf{U}} = \mathbf{H}\mathbf{U} + \mathbf{F},$$  \hspace{1cm} (1)

where $\mathbf{U}$ is a vector whose components are the unknown variables, $\mathbf{H}$ is an operator matrix containing the spatial derivatives and material properties, and $\mathbf{F}$ is the body force vector.

In (1) and elsewhere, time differentiation is indicated with the dot convention. The differential equation (1) correctly describes the anelastic effects in wave propagation within the framework of linear response theory. The solution of (1) subject to the initial condition

$$\mathbf{U}(t=0) = \mathbf{U}_0$$  \hspace{1cm} (2)

is formally given by

$$\mathbf{U}(t) = e^{\mathbf{H} t}\mathbf{U}_0 + \int_0^t e^{\mathbf{H} (t-\tau)} \mathbf{F}(\tau) d\tau.$$  \hspace{1cm} (3)

In equation (3), $e^{\mathbf{H} t}$ is called the evolution operator of the system. Solving (3) requires a suitable approximation for the spatial derivatives, which is achieved by the Fourier pseudospectral method [7]. Thus, equations (1), (2) and (3) should be replaced by the discretized equivalent equations.

The numerical solution is obtained by an optimal expansion of the evolution operator as polynomials, whose region of convergence depends on the spatial matrix $\mathbf{H}$, particularly on the location of its eigenvalues in the complex frequency plane. The form of $\mathbf{H}$ depends on the rheology and the unknown variables.

Let a plane wave solution to equation (1) be of the form

$$\mathbf{U} = \mathbf{U}_0 e^{i(\omega t - \mathbf{k} \cdot \mathbf{x})},$$  \hspace{1cm} (4)

where $\mathbf{x}$ is the position variable, $\omega_C$ is the complex frequency, and $\mathbf{k}$ is the real wave number vector. Substituting (4) into (1), and considering constant material properties and zero body forces, yields an eigenvalue equation for the eigenvalues $\omega = i\omega_C$. The determinant of the system must be zero in order for $\mathbf{U}_0$ to have a non-zero value. Therefore,

$$\det(\mathbf{H} - \mathbf{I}) = 0.$$  \hspace{1cm} (5)

where $\tilde{\mathbf{H}}$ is the spatial Fourier transform of $\mathbf{H}$, and $\mathbf{I}$ is the identity matrix. Hereafter, the complex plane of the eigenvalues is called the $\omega$-plane. Equation (5) determines the eigenvalues of $\mathbf{H}$ in the Fourier method approximation. Actually, the discretized equation should be used, but (5) represents a relative, good approximation.

The eigenvalues are analyzed in Section 2 for the following rheologies:

- **ANISOTROPIC VISCOELASTIC**
- **ISOTROPIC VISCOELASTIC**
The eigenvalue distribution defines the domain where the evolution operator is approximated by a suitable (rapidly converging) polynomial expansion. For each case, a brief review of the numerical integration techniques is given in Section 5. The methods are the following:

- Taylor expansion
- Chebyshev Spectral method
- Rapid expansion method
- Polynomial interpolation

through-conformal mapping
- Polynomial interpolation

by residum minimization

II. Wave Equations and Eigenvalues of \( \tilde{H} \)

Anisotropic-viscoelastic rheology

In order to implement Boltzmann's principle in the generalized Hooke's law, two relaxation functions based on the standard linear solid rheology are considered. One relaxation function describes the anelastic properties of the quasi-dilatational mode, and the other is related to the quasi-shear mode. This can be done by forcing the mean stress to depend on the first relaxation function, and the deviatoric components on the second (in this case, at least for some coordinate system, and usually along symmetry axes of the material). Moreover, the resulting rheological relation gives Hooke's law in the anisotropic-elastic limit, and the isotropic-viscoelastic rheology in the isotropic-anelastic limit [8], [5]. The equation of motion of a two-dimensional anisotropic-viscoelastic medium is formed with the following equations [1]:

i) The equation of momentum conservation:

\[
V \cdot \mathbf{T} = \rho \ddot{\mathbf{u}} + \mathbf{f},
\]

(6)

where \( \mathbf{T} = \{T_1, T_2, T_3, T_4, T_5, T_6\} \) is the stress vector, with \( \sigma_{ij} \), \( i,j = 1, \ldots, 3 \) the stress components. Defining the position vector by \( x = (x, y, z) \), \( u(x, t) \) and \( f(x, t) \) denote the displacement and body force vectors, respectively; \( \rho \) is the density, and \( \nabla \) in a divergence operator defined by

\[
V = - V_{ij} = \begin{bmatrix}
\partial \sigma_{ij}/\partial x & 0 & 0 & 0 & \partial \sigma_{ij}/\partial y & 0 \\
0 & \partial \sigma_{ij}/\partial y & 0 & \partial \sigma_{ij}/\partial z & 0 & \partial \sigma_{ij}/\partial x \\
0 & 0 & \partial \sigma_{ij}/\partial z & 0 & \partial \sigma_{ij}/\partial x & 0 \\
\end{bmatrix}.
\]

ii) The stress-strain relations:

\[
T_I = [A_{ij} + A_{ij}^{(0)}H_{ij3} + A_{ij}^{(1)} \sum_{l=1}^{L_v} \phi_l] \xi_{ij},
\]

(7)

where \( i,j = 1, \ldots, 6 \) and \( r = 1,2 \).

The methods for the numerical solution algorithm. For simplicity, a two-dimensional transversely-isotropic medium with symmetry axis parallel to the \( z \)-axis is considered. Then, \( c_{11}, c_{33}, c_{13} \) and \( c_{55} \) define the elastic characteristics of the medium. Choosing one relaxation mechanism for each mode \( (L_1 = L_2 = 1) \), the unknown variable vector is given by

\[
V^T = [u_x, u_y, \dot{u}_x, \dot{u}_y, \phi_1, \phi_2, \phi_3],
\]

(9)

where \( \phi_1 = c_{11}(1) + c_{11}(1), \phi_2 = c_{55}(2) - c_{55}(2), \) and \( \phi_3 = c(2) \), in terms of the memory variables. The spatial operator is

\[
H = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

where

- \( \rho H_{j1} = 0 \partial \phi_j/\partial x + \phi_j \partial \rho/\partial x \)
- \( \rho H_{j2} = 0 \partial \phi_j/\partial y + \phi_j \partial \rho/\partial y \)
- \( \rho H_{j3} = 0 \partial \phi_j/\partial z + \phi_j \partial \rho/\partial z \)
- \( \rho H_{j4} = 0 \partial \phi_j/\partial \xi_{12} + \phi_j \partial \rho/\partial \xi_{12} \)
- \( \rho H_{j5} = 0 \partial \phi_j/\partial \xi_{23} + \phi_j \partial \rho/\partial \xi_{23} \)
- \( \rho H_{j6} = 0 \partial \phi_j/\partial \xi_{31} + \phi_j \partial \rho/\partial \xi_{31} \)

\[
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

with

- \( \rho H_{j1} = 0 \partial \phi_j/\partial x + \phi_j \partial \rho/\partial x \)
- \( \rho H_{j2} = 0 \partial \phi_j/\partial y + \phi_j \partial \rho/\partial y \)
- \( \rho H_{j3} = 0 \partial \phi_j/\partial z + \phi_j \partial \rho/\partial z \)
- \( \rho H_{j4} = 0 \partial \phi_j/\partial \xi_{12} + \phi_j \partial \rho/\partial \xi_{12} \)
- \( \rho H_{j5} = 0 \partial \phi_j/\partial \xi_{23} + \phi_j \partial \rho/\partial \xi_{23} \)
- \( \rho H_{j6} = 0 \partial \phi_j/\partial \xi_{31} + \phi_j \partial \rho/\partial \xi_{31} \)

where

- \( H_{j1} = \phi_1 \partial \rho/\partial x, H_{j2} = \phi_2 \partial \rho/\partial y, H_{j3} = \phi_3 \partial \rho/\partial z, H_{j4} = \phi_4 \partial \rho/\partial \xi_{12}, H_{j5} = \phi_5 \partial \rho/\partial \xi_{23}, H_{j6} = \phi_6 \partial \rho/\partial \xi_{31} \).

907
where $D = (c_{11} + c_{33})/2$. The subindex $l$ denoting a physical mechanism has been omitted for simplicity. In the anisotropic-elastic limit, i.e., when $q^{(1)} = q^{(2)}$, and the memory variables vanish, equation (2.2) becomes Hooke's law. In the isotropic-viscoelastic limit, $c_{11} = \lambda + 2\mu$, $c_{13} = \lambda$ and $c_{33} = \mu$, with $\lambda$ and $\mu$ the Lamé constants, and (7) becomes the isotropic-viscoelastic rheology [5].

The eigenvalues of $\tilde{\gamma}$ are obtained from equation (5), where the following substitution: $\partial \partial x \rightarrow ik_x$, and $\partial \partial z \rightarrow ik_z$, with $k_x$ and $k_z$ the wavenumber components, gives $\tilde{\gamma}$ from $\gamma$.

The eigenvalue distribution for the different rheologies is displayed in Fig. 1. The material is a clayshale having $\gamma(1) = \gamma(2) = 0.0030 \, s^{-1}$, $\gamma(1) = 0.0027 \, s^{-1}$ and $\gamma(2) = 0.0025 \, s^{-1}$, which give highest dissipation around $f = 50 \, Hz$. The eigenvalues correspond to $k_x = k_z = 0.16 \, m^{-1}$. The negative real part of the propagating modes is a consequence of the anelasticity, stronger for the shear modes. The static modes arise from the fact that the formulation was done in the time domain; they are grouped approximately around $-1/(\tilde{\omega}(1))$ and $-1/(\tilde{\omega}(2))$. The differences are mainly due to anelasticity which introduces the static modes, since anisotropy only produces a shift of the wave mode eigenvalues in the vertical direction. Section 3 analyzes the appropriate methods for each rheology.

**Porous isotropic-viscoacoustic rheology**

Invoking the correspondence principle, Biot formally obtained a viscoelastic equation of motion which includes all possible dissipation mechanisms. The approach involves the presence of convolutional integrals which arise from the replacement of the elastic coefficients by time operators. When standard linear solid kernels are considered for the time operators, the equation of motion of the isotropic-viscoacoustic porous medium is given by the following equations [2]:

1) Biot equations:

$$v \left[ p_f \right] = \left[ \begin{array}{cc} -\rho_f & \rho_f \\ -\rho_f & m \end{array} \right] \left[ \begin{array}{c} \dot{u} \\ \dot{u} \end{array} \right] + \left[ \begin{array}{c} 0 \\ 0 \end{array} \right] \left[ \begin{array}{c} \eta_f \dot{k} \gamma \dot{u} \\ \eta_f \dot{k} \gamma \dot{u} \end{array} \right] + \left[ \begin{array}{c} s_f \\ s_f \end{array} \right].$$

(11)

where $p$ and $p_f$ are the pressure fields of the matrix-fluid system and fluid, respectively; $\dot{u}$ is the displacement of the solid; $u$ is a vector representing the flow of the fluid relative to the solid, and $s$ and $s_f$ are body force vectors. The material properties are: $\rho$, the composite density; $\rho_s$, the solid density; $\rho_f$, the fluid density; $m$, the tortuosity; $\eta$, the fluid viscosity; and $K$, the global permeability.
i) The stress-strain relations:

\[
\begin{bmatrix}
\psi_1 \\
\psi_2 \\
\psi_3
\end{bmatrix}
= \begin{bmatrix}
\psi_1 \\
\psi_2 \\
\psi_3
\end{bmatrix} + \sum_{i=1}^{m} \begin{bmatrix}
0 & 0 & 1 \\
0 & 1 & 0 \\
1 & 0 & 0
\end{bmatrix} \begin{bmatrix}
\phi_{21} \\
\phi_{21} \\
\phi_{31}
\end{bmatrix},
\]

where \( \phi \) and \( \xi \) are the dilatation fields of the solid matrix, and fluid relative to the solid, respectively; \( \psi_1, \psi_2, \psi_3 \) are memory variables. \( \psi_1 = (\rho + \rho_R + \rho_B) \phi \) and \( \psi_3 = \rho / \beta \), where \( \Delta, \rho_R, \) and \( \rho_B \) are the classical Biot elastic coefficients, and \( \beta \) is the porosity.

ii) The memory variable equations:

\[
\begin{bmatrix}
\phi_{11} \\
\phi_{51}
\end{bmatrix} = \begin{bmatrix}
\phi_{11} \\
\phi_{51}
\end{bmatrix} + \begin{bmatrix}
\phi_{11} \\
\phi_{51}
\end{bmatrix} - \frac{1}{c_0^2} \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
\phi_{11} \\
\phi_{51}
\end{bmatrix},
\]

for \( l = 1, \ldots, L \), where \( \phi_{11} = \phi_{51} = (1 - \rho / \rho_R) \phi_{11} \), \( \rho = 1, 3 \), with \( \tau_{11}^P \) and \( \tau_{11}^B \) relaxation times.

In the one-dimensional case with \( L = 1 \), the unknown vector \( V \) has nine components,

\[
V^T = [v, \xi, \dot{\xi}, -\ddot{u}, v_1, \xi_1, v_2, \xi_2].
\]

The spatial matrix \( H \) for constant material properties is given by

\[
H = \begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
H_{31} & H_{32} & 0 & 0 & H_{35} & H_{36} & H_{37} & H_{38} & H_{39} \\
H_{41} & H_{42} & 0 & 0 & H_{45} & H_{46} & H_{47} & H_{48} & H_{49} \\
H_{51} & H_{52} & 0 & 0 & H_{55} & H_{56} & H_{57} & H_{58} & H_{59} \\
H_{61} & 0 & 0 & 0 & 0 & 0 & H_{67} & 0 & 0 \\
H_{71} & 0 & 0 & 0 & 0 & 0 & 0 & H_{78} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & H_{99} & 0
\end{bmatrix}
\]

with \( \Delta = \delta^2 \partial x^2 \) and \( \gamma = \rho^2 - \rho m \). Biot poroelastic equations are obtained by taking \( \phi_{11} = \phi_P \), \( \rho = 1, 3 \). Then, the memory variables vanish and the unknown vector becomes \( V^T = [v, \xi, \dot{\xi}, -\ddot{u}] \). The equation for a viscoacoustic single-phase solid is obtained with \( \rho_1 = 0 \) and \( \phi_2 = \phi_3 = 0 \); only one set of relaxation times remains, corresponding to the solid phase \( \phi_1 \). The unknown vector in this case is \( V^T = [v, \xi, \dot{\xi}, -\ddot{u}] \).

Fig. 2. Eigenvalue distribution of \( \tilde{N} \) in the complex frequency plane for a porous viscoacoustic medium and limiting rheologies.
Substitution of $\partial\phi x$ by $ik$ gives the transformed matrix $\tilde{H}$. Fig. 2 shows the eigenvalue distribution. One of them is zero (not plotted) since the fourth and fifth rows of $\tilde{H}$ are linearly dependent. The slow modes present a quite diffusive behaviour due to the Biot mechanism. They are not present in the single-phase medium, whose attenuation characteristics are viscoelastic.

III. NUMERICAL INTEGRATION METHODS

To illustrate the different techniques, a zero source term is considered for simplicity in equation (1). A detailed formulation with source can be found in the respective references. The formal solution to the system is then given by

$$U(t) = e^{t\tilde{H}}U_0.$$  

(16)

The numerical solution for general inhomogeneous media requires a polynomial representation of the evolution operator. The different methods are:

Taylor expansion A Taylor expansion of the evolution operator up to the second order is

$$e^{t\tilde{H}} \approx 1 + Ht + \frac{1}{2} H^2 t^2.$$  

(17)

Replacing (17) into (16), and substracting $U(-t)$ from $U(t)$ gives

$$U(t) = U(-t) + 2t HU_0.$$  

(18)

This formula basically gives the equations for second-order temporal differencing valid for small $t$ [7]. Although the region of convergence of the Taylor expansion is the whole $z$-plane, in order to have high accuracy, the time step should be very small; more precisely, $\Delta t = \Omega(N^{-2})$, using finite-order explicit schemes, where $N$ is the number of grid points.

Chebychev spectral method This technique makes use of the following expansion of $e^z$ [11]:

$$e^z \simeq \sum_{k=0}^{K} C_k J_k(zR)Q_k \left[ \frac{z}{z_{R}} \right],$$  

(19)

where $|z| \leq tr$, and $z$ lies close to the imaginary axis. $C_k = 1$ and $C_k = 2$ for $k \geq 1$, $J_k$ is the Bessel function of order $k$, and $Q_k$ are modified Chebychev polynomials which satisfy the recurrence relation

$$Q_{k+1}(z) = 2zQ_k(z) + Q_{k-1}(z), \quad Q_0 = 1, \quad Q_1 = z.$$  

(20)

Substituting $\tilde{H}$ for $z$ in (19), equation (16) becomes

$$U(t) \cong \sum_{k=1}^{K} C_k J_k(zR)Q_k \left[ \frac{z}{z_{R}} \right] U_0.$$  

(21)

The series has a rapid convergence for $K > tr$, with $K = O(N)$. The value of $R$ should be chosen larger than the range of the eigenvalues of $\tilde{H}$. Since this expansion converges for the imaginary axis of the $z$-plane, it is appropriate for the elastic case [7]. Anelastic problems can be solved with less efficiency using a slight modification [4].

Rapid expansion method In the elastic case where no first time derivatives of the displacements and memory variables are present, the wave equation of the system can be expressed as

$$\ddot{u} = -L^2 u + f,$$  

(22)

where $u$ is the displacement vector, $f$ is the body force vector, and $-L^2$ is a linear matrix operator similar to $\tilde{H}$ [9]. For zero body forces the formal solution to (22) is

$$u(t) = \cos Lt u(0) + \frac{\sin Lt}{L} \dot{u}(0).$$  

(23)

Adding solutions (23) for times $t$ and $-t$, the displacement time derivative can be eliminated, and the displacement at time $t$ becomes

$$u(t) = u(-t) + 2 \cos(Lt) u(0).$$  

(24)

The method uses the following expansion:

$$K\sum_{k=0}^{\infty} C_2KJ_2k(tR)Q_2k \left[ \frac{11}{R} \right],$$  

(25)

This expansion represents an improvement over the Chebychev spectral method since it contains only even order functions $Q_{2k}$, however, it can be used only for elastic problems [6].

Polynomial interpolation through conformal mapping As shown in the previous section, in a single phase anelastic solid, the eigenvalues of $\tilde{H}$ lie on a T-shaped domain $D$ which includes the negative real axis and the imaginary axis. This approach is based on a polynomial interpolation of the exponential function in the complex domain $D$, on a set of points which is known to have maximal properties. This set, known as Fejer points, is found through a conformal mapping between the unit disc and the domain of the eigenvalues $D$. In this way, the interpolating polynomial is "almost best" [10].

Getting the Fejer points is as follows: Let $\chi(u)$ be a conformal mapping from the $u$-plane to $z$-space, which maps the complement of a disc of radius $\delta$ to the complement of $D$, where $\delta$ is the logarithmic capacity of $D$, given by the limit $\delta = [\chi'(\infty)]$, the prime denoting derivative with respect to the argument. The analytic expression for $\chi(u)$ corresponding to the domain $D$ can be found in [10]. The same function $\chi(z)$ maps the complement of the unit disc to the complement of the domain $D$.

Then, the Fejer points are $z_j = \chi(u_j)$, $j = 0, \ldots, m-1$ where $u_j$ are the $m$ roots of the equation $\omega^m = \delta$, with $m$ the degree of the polynomial. The set $[z_j]$, $j = 0, \ldots, m-1$ has maximal properties of convergence. Then, the sequence of polynomials $P_m(z)$ of degree $m$ found by interpolation to an arbitrary function $f(z)$, analytic on $D$ at the points $z_j$, converge maximally to $f(z)$ on $D$. The interpolating polynomial in Newton form is

$$P_m(z) = a_0 + a_1(z - z_0) + a_2(z - z_0)(z - z_1) + \ldots + a_m(z - z_0)\ldots(z - z_{m-1}),$$  

(26)

where $a_j = I(z_0, \ldots, z_j)$, $j = 0, \ldots, m-1$. are the divided differences. The approximating polynomial is given by $P_m(\tilde{H})$ with $f(z) = e^z$.  

910
Polynomial interpolation by residuum minimization. The preceding method requires a conformal mapping from the unit disc to the domain $D$, with the eigenvalues of $H$ to find the interpolating points. This new technique avoids the conformal mapping by finding the interpolating points automatically in an optimal way [12]. Therefore, the method can be applied for any general matrix $H$, no matter what the domain $D$.

The idea is to find the interpolating points by minimizing the $L^2$-norm of the error. It is well known that the error of the interpolation is

$$E_m(z) = f(z) - P_m(z) = \frac{f^{(m)}(z)}{m!} R_m(z),$$

with

$$R_m(z) = \prod_{i=1}^{m} (z - z_{i-1}) = \sum_{k=0}^{m-1} a_k z^k + z^m$$

and $s$ the value for which $f(z) - P_m(z) - E(z) = 0$. The superindex $(m)$ denotes the $(m)^{th}$ derivative. Substituting $H$ for $z$ in (27a) and using (16), the error of the algorithm is

$$E_m = \frac{f^{(m)}(z)}{m!} \Sigma_m,$$

where $\Sigma_m = R_m(Ht)u_0$. (28a - b)

Minimizing the $L^2$-norm $\|\Sigma_m\|^2 = (\Sigma_m, \Sigma_m)$ is achieved by solving the following set of $m$ linear equations:

$$\frac{\partial}{\partial H(i)} \|\Sigma_m\|^2 = 0, \quad i = 0, \ldots, m - 1.$$ (29)

This is equivalent to solve the following system:

$$DA = B, \quad A = [e_0, \ldots, e_{m-1}]^T.$$ (30)

where

$$D_{ij} = (Ht)^{i-1}u_0, \quad B_i = -(Ht)^{i-1}u_0, \quad Ht^m u_0,$$

$$1 \leq i, j \leq m.$$ (31a, 31b)

After solving for $A$, the interpolating points are obtained from the roots of $R_m(z)$. The approximating polynomial is given by (26) with $f(z) = z^2$. Further research is required to determine whether this technique improves the efficiency when solving anelastic wave propagation problems.

IV. CONCLUSIONS

This work briefly reviews some of the theories and algorithms for solving wave propagation problems in linear viscoelastic media. The methods use spectral techniques and solve the wave equation in the time-domain. A consistent introduction of Boltzmann's after-effect principle in the time-domain, for anisotropic and porous media, is achieved by the introduction of memory variables. Some additional assumptions are required in the anisotropic case for the determination of the constitutive relations. The eigenvalue analysis for each rheology indicates that spectral Chebyshev methods are suitable for viscoelastic problems, and that polynomial interpolation techniques are required when the medium is anelastic.

REFERENCES


Abstract - This paper deals with a new method for some heat transfer problems through a system of walls and caves in buildings, the caves being ventilated and heated. From an energy balance argument in the caves, the outgoing air at every time instant formulation of the problem, taking into account the non perfect thermal contact conditions on both sides of the interfaces, the heat transfer through the cavity structure is a weak solution may be shown. The resulting recurrent problem, with Ventilation and heating lead to non standard transition conditions involving Volterra operators, acting on the traces of the temperature from both sides of the interfaces.

Crucial in the analysis is a non standard variational formulation of the problem, taking into account the non perfect thermal contact conditions in an appropriate way. By the method of discretization in time, see e.g. (1), the existence of a unique, stable weak solution may be shown. The resulting recurrent system of elliptic problems at each subsequent time point is approximated by a FEM. Both the convergence and the error estimates for the semi-discrete and fully discrete approximation scheme are stated. In a simple case the numerical results show a good agreement between the exact and the approximate solution.

1. INTRODUCTION.

To avoid distracting technicalities, we confine ourselves to a 1-D model problem with practical relevance viz the simple cavity structure of Fig. 1., see e.g.

\[\begin{align*}
\text{wall 1} & \quad \text{cave} & \quad \text{wall 2} \\
0 & \quad \Omega_1 & \quad \Omega_2 & \quad x = \xi_1, \xi_2
\end{align*}\]

Fig. 1. Cross section of a simple cavity structure

The air space, assumed to be homogeneously at temperature \(T(t)\), is contained between two walls (with conductivities \(k^1\) and diffusivities \(k^1\)), being at temperatures \(u^1, 1 = 1, 2\). At \(x = \xi_1\) and \(x = \xi_2\) heat is transferred by convection between the wall surfaces and the air, with respective transmission coefficients \(h^{1,2}\) and \(h^{2,1}\). The radiative heat transfer between these surfaces is linearized, with coefficients \(h^{1,2} = h^{2,1}\).

The walls consist of two parallel isotropic slabs, assumed to conduct heat only in one direction, orthogonal to the surfaces. As we focus on the contact problem, we take the surfaces \(x = 0\) and \(x = \xi_3\) to be insulated, for simplicity in the formulation. However, inhomogeneous Neumann and Dirichlet or Robin conditions can be covered as well by the present approach.

The cave is heated homogeneously at a rate \(q(t)\). Finally the cave is ventilated, both directly by incoming air with velocity \(v(t)\) and temperature \(T_0(t)\), and indirectly by a change of air with rate \(K\) per unit time interval per unit temperature difference, the outdoor air temperature being \(0(t)\). In modelling the ventilation we assume that:

- the incoming air is mixed very quickly with the air in the cave
- the mass of the incoming air equals to the mass of the outgoing air at every time
- the temperature and convective transfer are uniform over the wall surfaces \(x = \xi_1\) and \(x = \xi_2\).

2. MATHEMATICAL MODEL

Under standard assumptions the mathematical problem of the heat transfer through the cavity structure is:

\[u^1(x,t) \quad \text{s.t.} \quad k^1 \frac{\partial u^1}{\partial x} - \frac{\partial}{\partial x} \left( k^1(x,t) \frac{\partial u^1}{\partial x} \right) = 0, \quad \forall x \in \Omega_1, \quad t > 0, \quad \text{with boundary conditions} \]

\[\frac{\partial u^1}{\partial x}(x,t) = 0, \quad t > 0, \quad \frac{\partial u^1}{\partial x}(x,t) = 0, \quad t > 0 \quad \text{initial conditions} \]

\[u^1(x,0) = u^1_0(x) \quad \forall x \in \Omega_1, \quad t > 0 \quad \text{where} \]

\[g_1 = S_1/V, \quad g_2 = S_2/V \quad \text{where} \]

\[\text{Consequently,} \quad T(t) = c^\ast \left( k^1 \cdot T_0 + \frac{1}{c_p} \cdot q(t) \right) \quad \text{where} \]

\[\begin{align*}
&\left( g_1 \cdot h^{1,2} \cdot u^1(t,s) + h^{2,1} \cdot u^2(t,s) + K \cdot q(t) \right) \\
&+ K \cdot q(t) + g_2 \cdot c_p \cdot \frac{\partial (T_0 + q(s))}{\partial s} + G(u^1, u^2)(t)
\end{align*}\]
\[ \kappa(t) = \int \left( (g_1/c_p) \cdot (h^{1/2} + h^{2/1}) + g_2 \cdot v + K/c_p \right) \, ds. \]

Substituting this expression in the transition conditions (2.3)–(2.4) and translating the interval \((\xi_2, \xi_3)\) to \((\xi_1, \xi_0)\), with \( L = \xi_2 - \xi_3 \), we arrive at a parabolic problem for the functions \( u(x, t) \), \( x \in \Omega \), \( t > 0 \), \( 1 = 1.2 \) of the type mentioned above. This is a problem in a two-component medium with a jump both in the flux and in the temperature at the interface \( x = \xi_1 \) of the 2 subregions.

**Remark 2.1.** In practice, when \( T(t) \) varies slowly in time, the left hand side of (2.6) may be neglected, \( t_p \) being small (in popular models, even \( c_p = 0 \), \( \frac{\pi}{4} \)). Then \( T(t) = G(u^1, u^2)(t) \) takes a particular simple form.

The analysis may easily be extended to a structure with \( H \) parallel walls and \( H \)-I enclosed caves (\( H > 2 \)).

### 3. Variational Formulation

#### 3.1. Notations and Assumptions

Let \( H^1(\Omega) \) be the usual first order Sobolev space on \( \Omega \), with norm \( \| \cdot \|_1 \), \( 1 \leq 1 \leq 2 \). We set

\[ V = \{ u = (u^1, u^2) \mid u^1 \in H^1(\Omega_1), 1 \leq 1 \leq 2 \} \]

and identify \( u \in V \) with the scalar function \( u : \Omega \to \mathbb{R} \) for which \( u|_{\Omega_1} = u^1 \) on \( \Omega_1 \), \( 1 \leq 1 \leq 2 \). Similarly we deal with the product space \( H = L^2(\Omega_1) \times L^2(\Omega_2) \). Denote

\[ \begin{align*}
(u, v)_H = \sum_{i=1}^{2} \int_{\Omega_1} u^i \cdot v^i \, dx, & \quad \text{b}(t; u, v) = \sum_{i=1}^{2} \int_{\Omega_1} u^i \cdot v^i \, dx, \\
|u|_H^2 = \sum_{i=1}^{2} \int_{\Omega_1} |u^i|^2 \, dx, & \quad \|u\|_V^2 = \left( \sum_{i=1}^{2} \|u^i\|_2^2 \right)^{1/2},
\end{align*} \]

\[ V_u, v \in V \]

Let \( T > 0 \) be a given number and set \( I = (0, T) \). We use the standard functional spaces \( C(I, X), L^2(I, X), \)

\[ L^2(\Omega, X), \text{ etc.} \]

where \( X \) is a Banach space.

Throughout we make the assumptions

\[ u^1, \, k^1 \in \text{Lip}(I, L^2(\Omega_1)); \quad u^2, \, k^2 \in (p \text{ constant}), \quad 1 \leq 1 \leq 2 \]

\[ h^{1/2}, \, h^{2/1} \text{ and } h^{2/2} = h^{3/1} \text{ in Lip}(I, \mathbb{R}^+) \]

\[ v(t), \, 0(t), \, q(t) \in \text{Lip}(I, \mathbb{R}) \]

\[ u^1 \in H^1(\Omega_1), \quad 1 \leq 1 \leq 2. \]

#### 3.2. Variational Problem

**Definition 3.1.** A function \( u : I \to \Omega \), with \( u \in L^p(I, \Omega) \) and \( \partial u \in L^1(I, \Omega) \), is a variational solution of (2.1)–(2.5), (2.7). In the time interval \( I \), if

\[ b(t; \partial u(t), \phi) + a(t; u(t), \phi) \]

\[ + h^{1/2} \{ u'(t, \xi_1) \} - G(u^1, u^2)(t) \} \cdot \phi'(t_1) \]

\[ + h^{3/2} \{ u'(t, \xi_1) \} - G(u^1, u^2)(t) \} \cdot \phi'(t_1) \]

\[ + h^{2/2} \{ u'(t, \xi_1) \} - u^2(t_1, \xi_1) \} \cdot \phi'(t_1) - \phi'(t_1) \]

\[ = 0 \]

\[ \forall \phi \in V. \quad \text{a.e. in } I \]

\[ u(0) = u_0 \]

From the smoothness of the data and the required regularity of \( u \), all integrals in (3.2) exist.

The relation (3.2) is obtained from (2.1)–(2.4) by first dealing with the problems for \( u^1 \) and \( u^2 \) in the usual way and by next adding the resulting variational equations, taking into account the notations (3.1). The formal equivalence of the classical and the weak variational problem can readily be proved.

### 4. Discretization in Time

Consider \( n \in \mathbb{N} \), a time step \( \Delta t = T/n \) and time points \( \tau_j = j \Delta t, 0 \leq j \leq n \). We define \( \tilde{u}_j^1 \in V \), intended to be an approximation of \( u(x, \tau_j) \), \( 1 \leq j \leq n \), by the linear recurrent system

\[ b(t_j; \delta u^1_j, \phi) + a(t_j; u^1_j, \phi) \]

\[ + h^{1/2} \{ u_j^1(t_1) \} - G(u_j^1, u_j^2)(t_j) \} \cdot \phi'(t_1) \]

\[ + h^{3/2} \{ u_j^1(t_1) \} - G(u_j^1, u_j^2)(t_j) \} \cdot \phi'(t_1) \]

\[ + h^{2/2} \{ u_j^1(t_1) \} - u_j^2(t_1) \} \cdot \phi'(t_1) - \phi'(t_1) \]

\[ = 0 \]

\[ \forall \phi \in V. \quad 1 \leq j \leq n \]

\[ (4.1) \]

where

\[ \delta u^1_j = (u_j^1 - u_{j-1}) / \Delta t, \quad h^{3/2} = h^{1/2} \{ t_j \}, \text{ etc.} \]

\[ \tilde{u}^1_j(t_j) = \left\{ \begin{array}{ll}
\tilde{u}^1_j & \text{for } t \in (t_j-1, t_j), \quad 1 \leq j \leq n-1, \\
\tilde{u}^1_j & \text{for } t \in (t_{j-1}, t_j-1). \end{array} \right. \]

By the Lax-Hilbert lemma it can be shown that the (elliptic) problem (4.1) for \( \tilde{u}^1_j \in V \) has a unique solution in terms of \( u_j^1 \), \( j = 0, \ldots, n-1 \).

**Definition 4.1.** The Rothe function \( u^{(n)} : I \to \mathbb{V} \), intended to be an approximation of \( u \), is introduced by \( u^{(n)}(t) = u_j^1 + \delta u^1_j (t - t_j), \quad \tau_j \leq t \leq \tau_j+1 \), \( 1 \leq j \leq n \).

\[ (4.2) \]

From here on \( C > 0 \) is a generic constant neither depending on \( \Delta t \) nor \( \ln \Delta t \) on \( \lambda \).

Proceeding to some extent similarly as in [21],[3], we may prove

**Theorem 4.1.** There exists a function \( u \in C(I, \mathbb{H}) \cap L^2(I, \mathbb{V}) \), with \( \partial u \in L^2(I, \mathbb{H}) \) (if \( u \) is differentiable a.e. in \( I \)) such that \( u^{(n)} \to u \) in \( C(I, \mathbb{H}) \cap L^2(I, \mathbb{V}) \) and \( \partial u \) \( a \to u \) in \( L^2(I, \mathbb{H}) \) for \( n \to a \). Moreover

\[ \| u - u^{(n)} \|_{C(I, \mathbb{H})} + \| u - u^{(n)} \|_{L^2(I, \mathbb{V})} \leq C/\sqrt{n} \]

Finally \( u \) is the solution in the sense of Defin. 1. 1.

**Remark 4.1.** When the initial function \( u_0 \in V \) satisfies a "compatibility condition" viz. when there exists \( \zeta_0 \in \mathbb{H} \), to be interpreted as \( \delta u^0 \) such that (4.1) holds for \( j = 0 \) too, then the estimate (4.3) can be improved to \( O(1/n) \). Moreover, then \( u \) \( \in \text{Lip}_{1/2}(I, \mathbb{V}) \) and \( \partial u \in L^2(I, \mathbb{H}) \cap L^2(I, \mathbb{H}) \).

### 5. Full Discretization

#### 5.1. Abstract error estimate

Consider a family \((V^h, \lambda, \Lambda)\) of (finite element) subspaces of \( V \). Introduce \( \Lambda^h \in V^h \), the Galerkin approximation of \( u \), \( 0 \leq j \leq n \), by a similar recurrent system
to (4.1), where now \( p \in \mathcal{V}_\lambda \) and where \( u_0 \) is replaced by a suitably chosen approximation \( u_0^\lambda \in \mathcal{V}_\lambda \).

Put \( \alpha = (\Delta t, \lambda) \). The discrete Rothe function \( u_0(\alpha) \) is defined by a similar relation to (4.2). To obtain the counterpart of Theorem 4.1, we assume

\[
\| u_0^\lambda - u_0 \|_V \text{ if } \lambda \to 0
\]

(5.1)

For any \( z \in L_2(I, V) \) there exists \( z_\lambda \in L_2(I, V_\lambda) \) such that \( z_\lambda \to z \) in \( L_2(I, V) \) if \( \lambda \to 0 \)

(5.2)

For finite element choices of \( V_\lambda \subset V \), (5.2) is implied by an inequality of the type (5.5) below.

**THEOREM 5.1.** Under the additional assumptions (5.1)-(5.2) the convergences, mentioned in Theorem 4.1, are valid for \( u_0(\alpha) \) if \( \alpha \to 0 \). Moreover,

\[
\| u_{\alpha} - u \|_{C(I, H)}^2 + \| u_{\alpha} - u \|_{L_2(I', V)}^2
\]

\[
\leq C(\Delta t + \| u_0 \|_H^2 + \| u \|_{L_2(I', V)}^2)
\]

(5.3)

**Remark 5.1.** This estimate can be improved, viz. \( \Delta t \) may be replaced by \((\Delta t)^2\), when \( u_0^\lambda \) satisfies a 'compatibility relation'. More precisely \( u_0^\lambda \in \mathcal{V}_\lambda \), defined by a similar relation to (4.1) with \( j = 0 \), \( \alpha \in \mathcal{V}_\lambda \), and \( u_0 \) replaced by \( u_0^\lambda \), has to obey \( \| u_0^\lambda \|_H^2 \leq C \) if \( \lambda \to 0 \).

5.2. Rate of convergence in the mesh parameter

Let \( (\eta^1)_{\lambda=0} \) be a regular family of partitions of \( \Omega_t \), \( 1 \leq 1 \leq 2 \), with global mesh parameter \( \lambda \). Introduce

\[
X_{\lambda} = \{ v^1 \in C^0(\tilde{\Omega}_\lambda) \mid v^1|_{K} \in P^1(K) \quad \forall K \in t^1_{\lambda} \}
\]

\[
V_{\lambda} = \{ v = (v^1, v^2) \mid v^1 \in X_{\lambda} \quad 1 \leq 1 \leq 2 \} \subset V
\]

(5.4)

For the Lagrange finite element spaces \( X_{\lambda} \) we know

\[
\| v \|_{X_{\lambda}}^2 - \| v \|_{X_{\lambda}}^2 \leq \lambda \| v \|_{X_{\lambda}}^2 \leq C \lambda^2 \| v \|_{X_{\lambda}}^2
\]

(5.5)

where \( v^1 \) is the standard Lagrange interpolator or its Clement's generalization to \( L_2(\Omega_t) \), \( 1 \leq 1 \leq 2 \).

Moreover, taking \( u_0^\lambda = (\pi^1_{\lambda}, u^1_{\lambda}, \pi_\lambda^2, u^2_{\lambda}) \), we get

\[
\| u_0^\lambda - u_0 \|_H^2 \leq C \lambda^2 \| u_0 \|_H^2
\]

Combining these estimates with Theorem 5.1, we have

**THEOREM 5.2.** Assume that \( u^1 \in L_2(I, H^2(\Omega_t)) \), \( 1 \leq 1 \leq 2 \). Take \( \mathcal{V}_\lambda \subset V \) and \( u_0^\lambda \in \mathcal{V}_\lambda \) as indicated above, then

\[
\| u_{\alpha} - u \|_{C(I, H)}^2 + \| u_{\alpha} - u \|_{L_2(I', V)}^2 = O(\Delta t + \lambda^2)
\]

If the compatibility condition on \( u_0^\lambda \), mentioned in Remark 5.1, is satisfied, then this estimate is improved to \( O((\Delta t)^2 + \lambda^2) \).

6. NUMERICAL EXAMPLE

We consider a simple test problem, cfr. Remark 2.1, the exact solution of which is known:

Determine \( u^1 \) in \((0, 1)\) and \( u^2 \) in \((1, 2)\), \( t > 0 \), obeying the D.E.'s (2.1), where \( k^1 = 1, \lambda^2 = 2, u^1 = u^2 = 1 \), together with the B.C.'s (2.2) at \( x = 0 \) and \( x = 2 \)

respectively, as well as with the T.C.'s at \( x = 1 \)

\[
- \delta \Delta u^1 = 5 \times (u^1 - u^2) + 5, \quad 2 \times \delta \Delta u^2 = 5 \times (u^2 - u^1) - 3
\]

and the I.C.'s

\[
u^1(x, 0) = \cos(\beta x, \cdot x), \quad u^2(x, 0) = (-1/\sqrt{2}) \sin(\beta x, \sqrt{2}) \cos((\beta/\sqrt{2}) \cdot (x-2))
\]

Here \( \beta = 1.590724 \) is the first positive root of

\[
\cos(\beta/\sqrt{2}) \cdot \cos(\beta/\sqrt{2}) = \beta/5
\]

By the method of seperation of variables the analytical solution reads

\[
u^1 (x, t) = e^{-\beta^2 \cdot t}, \quad u^2 (x, t) = e^{-\beta^2 \cdot t}, \quad u^2 (x, t) + x
\]

As in § 5.2 we use a linear FEM with nodes

\[
x^\ell = \delta_{I-1,1} + \ell \lambda, \quad 0 \leq \ell \leq \frac{1}{\lambda} \in \mathbb{N}_0, \quad 1 \leq 1 \leq 2
\]

As error characteristics we use discrete \( L_1 \) and \( L_\infty \) norms

\[
e_1 (t) = \sum_{I=1}^{1/\lambda} \| u(x^\ell, t) - u_\alpha (x^\ell, t) \|_I
\]

\[
e_1 (h) = 100 \cdot e_1 (t)/\sum_{I=1}^{1/\lambda} \| u(x^\ell, t) - u_\alpha (x^\ell, t) \|_I
\]

(5.10)

Hence, the table below shows accurate results, even for a coarse mesh.

\[
\left| \frac{\Delta t}{10} \cdot e_1 (t) \right|
\]

\[
\left| \frac{\Delta t}{10} \cdot e_1 (h) \right|
\]

<table>
<thead>
<tr>
<th>( t )</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( e_1 (t) )</td>
<td>0.04</td>
<td>0.09</td>
<td>0.14</td>
<td>0.19</td>
<td>0.2</td>
</tr>
<tr>
<td>( e_1 (h) )</td>
<td>0.002</td>
<td>0.0038</td>
<td>0.0049</td>
<td>0.005</td>
<td>0.006</td>
</tr>
</tbody>
</table>

Table. Case I : \( \Delta t = 10^{-2}, \lambda = 1/4 \).
Case II : \( \Delta t = 10^{-4}, \lambda = 1/32 \).

Using a Crank-Nicholson modification of the semi-discrete scheme of § 4, the results may be improved by about 30%. This and other generalizations, e.g. to 3D problems including non-linear phenomena, are investigated in a forthcoming paper.

REFERENCES

Abstract—A linearised cylindrical wave motion is considered for a fluid of finite depth in the presence of an impermeable cylindrical wall and coaxial porous wall immersed vertically in the fluid. The motion is generated once by the wavemaker immersed in water was solved by Havelock [1], Biesel & Suquet [2] and Ursell, Dean & Tu [3]. In these works, the wavemaker generated by a harmonically oscillating vertical and coaxial porous wall immersed vertically in the presence of an impermeable cylindrical wall and suitably limited. A coaxial cylindrical porous wall of circular cross-section of radius b is considered.

1. Introduction

The classical problem of forced two-dimensional wave motion with outgoing wave at infinity generated by a harmonically oscillating vertical wavemaker immersed in water was solved by Havelock [1], Biesel & Suquet [2] and Ursell, Dean & Tu [3]. In these works, the wavemaker was represented by a vertical impermeable plate immersed vertically in a fluid.

2. Waves generated by the impermeable wall

We are concerned with the irrotational motion of fluid with free surface which is assumed to be incompressible and inviscid, flow under the action of gravity. The motion is induced by an impermeable vertical cylindrical wall of circular cross-section of radius a. The wall assumed to perform radial harmonic oscillations normal to its axis, let its velocity at time t be $U(y) \exp(-i\omega t)$, where $U(y)$ is a complex valued, and suitably limited. A coaxial cylindrical porous wall of circular cross-section of radius b is fixed in the fluid. The resulting motion on is therefore axisymmetric and time harmonic with the same angular frequency $\omega$ as that of the porous wall.

Let $(r,y)$ be cylindrical polar coordinates with the origin 0 in the undisturbed free surface such that by pointing down into the fluid coinciding with the axis of the porous wall. We consider the case when the fluid is of finite depth. Let $\phi_j$ be the velocity potentials where the subscripts j=1,2 refer to the regions $a < r < b$, $r > b$ and the functions $\phi_j$ satisfy

$$\frac{\partial^2 \phi_j}{\partial r^2} + \frac{1}{r} \frac{\partial \phi_j}{\partial r} + \frac{\partial^2 \phi_j}{\partial y^2} = 0, \quad y > 0$$

$K \phi_j + \frac{\partial \phi_j}{\partial y} = 0, \quad y = 0, \quad K = \omega^2 / g$ (2.2)

$$\frac{\delta \phi_j}{\delta r} = U(y), \quad y > a$$

$$\frac{\delta \phi_j}{\delta r} = \frac{\partial \phi_j}{\partial r}, \quad y = b$$

We shall assume that the porous wall is made of material with very fine pores. Thus according to Bvrcy's law [5,6], we have

$$\frac{\partial^2 \phi_j}{\partial r^2} + \frac{1}{r} \frac{\partial \phi_j}{\partial r} + \frac{\partial^2 \phi_j}{\partial y^2} = 0, \quad y = h, \quad \mu$$

is the dynamic viscosity, $\rho$ is the density and $b$ is a coefficient which has the dimension of length.
when the quantity \( x \) the new problem is the le the impermeable wall \( r = a \) be kept fixed then the ratio of the amplitude of the reflected
diffrent from zero. However, it indicates that \( C H_0(K r) \cosh k (h - y) \), \( \phi_2 = 0 \), as expected. This solution is valid only when the quantity \( J_1(k b) Y_1(k a) - J_1(k a) Y_1(k b) \) is different from zero. However, it indicates that when this quantity vanishes, resonance occurs and the linearised theory can not be applied.

4. Waves generated by the porous wall

If we let now the porous wall oscillate radially about \( r = a \), with velocity \( U(y) \exp(-1wt) \) while the impermeable wall \( r = a \) be kept fixed, then the new problem is the same as stated in section 2 except that (2.3), (2.5) are replaced by

\[
\begin{align*}
\frac{\partial \phi_1}{\partial r} &= 0, \quad \text{on } r = a \\
\frac{\partial \phi_2}{\partial r} - U(y) &= iG(\phi_1 - \phi_2), \quad \text{on } r = b,
\end{align*}
\]

In this case we get

\[
\begin{align*}
\phi_1 &= \frac{2i\kappa b a \cos k h K_1(k b)}{\delta_0(\delta k)} \cos k h (h - y) + I_0(k b) K_1(k a) + K_0(k r) I_1(k a) = k b a \cos k h (h - y) + I_0(k b) K_1(k a), \\
\phi_2 &= \frac{2i\kappa b a \cos k h K_1(k b)}{\delta_0(\delta k)} \cos k h (h - y) + I_0(k b) K_1(k a) + K_0(k r) I_1(k a) = k b a \cos k h (h - y) + I_0(k b) K_1(k a),
\end{align*}
\]

and

\[
\psi_2 + (A-C) H_1^{(1)}(kr) \cosh k(h-y) = r \to \infty \quad (5.8)
\]

Since the present problem is linear, \( \psi_1, \psi_2 \) can be obtained by suitable superposition of the results (3.3), (3.4) and (4.3), (4.4) respectively. Hence

\[
\begin{align*}
\phi_1 &= \frac{2G}{\delta k} H_1^{(1)}(kr) - J_1(k a) H_0^{(1)}(kr) \cosh k (h - y), \\
\phi_2 &= -G \Delta ' (k) H_0^{(1)}(kr) + C H_0^{(2)}(kr) \cosh k (h - y)
\end{align*}
\]

where \( \Delta (k) = \frac{G}{\delta k} H_1^{(1)}(kr) - J_1(k a) H_0^{(1)}(kr) \) and \( \phi_2 = 0 \). The coefficient of reflection \( R \) is defined as the ratio of the amplitude of the reflected wave to the amplitude of the incident wave

\[
R = \frac{\Delta ' (k)}{\Delta (k)} = \frac{G}{\delta k H_1^{(1)}(kr) - J_1(k a) H_0^{(1)}(kr)}, \quad 0 < \Delta (k) \leq 1 \quad (5.9)
\]

where

\[
\alpha^2 = \frac{4n b k}{J_1(k a) + J_1(k b)}, \quad \beta^2 = \frac{4n b k}{J_1(k b) + J_1(k a)}
\]

This solution is valid only when the wall is completely permeable but now the wave is totally reflected at the impermeable wall at \( r = a \). We note also that when \( 0 \leq \alpha \leq 1 \), the incident wave is totally reflected irrespective of the value of \( \alpha \) by simple differentiation of (5.9), we note that the value of \( R \) for any fixed \( a \) and \( b \) reduces to a minimum

\[
R = \min \left[ \frac{\alpha - \beta}{\alpha + \beta + M} \right],
\]

when \( \frac{\alpha - \beta}{\alpha + \beta + M} \), this minimum value vanishes when \( \alpha = \beta = M \) or when a and b satisfy the equation

\[
J_1(k b) Y_1(k a) - J_1(k a) Y_1(k b) = 0 \quad (5.10)
\]

in this case \( \theta = \beta \). Under these circumstances the porous wall acts as an efficient wave absorber or eliminator for the incident waves, i.e. for the values of \( G/\kappa b \alpha \) and a and b satisfy equation (5.10), there is a wave trapping phenomenon that as wave will be trapped inside the region \( a \leq r \leq b \).

References

Abstract - Steady plane magnetohydrodynamic flow of a viscous incompressible fluid of infinite electrical conductivity is governed by

\[
\begin{align*}
\text{div} \mathbf{V} &= 0 \\
\rho (\mathbf{V} \cdot \text{grad}) \mathbf{V} + \mu \text{grad}^2 \mathbf{V} + \mu (\text{curl} \mathbf{H}) \times \mathbf{H} &= \mathbf{0} \\
\text{curl} (\mathbf{V} \times \mathbf{H}) &= 0 \\
\text{div} \mathbf{H} &= 0
\end{align*}
\]

where \( \mathbf{V} = (u, v) \) denotes the velocity vector, \( \mathbf{H} = (H_x, H_y) \) the magnetic field vector, \( \rho \) the pressure, \( \mu \) the constant fluid density, \( \mu \) the constant coefficient of viscosity and \( \mu \) the constant magnetic permeability. The magnetic field vector \( \mathbf{H} \) is given by the solution of

\[
\text{curl} (\mathbf{V} \times \mathbf{H}) = 0
\]

when the magnetic field is orthogonal to the velocity field. Writing the governing equations in \( x, y \) coordinates and recasting these in new independent variables \( z = x + iy \) and \( \bar{z} = x - iy \), we find that:

If \( \psi(z, \bar{z}) \) is the streamfunction of the flow, then \( \psi(z, \bar{z}) \) must satisfy

\[
\begin{align*}
\frac{1}{4 \pi} \text{Im} \left\{ \psi \frac{1}{z \bar{z}} \right\} &= \frac{1}{2 \mu} \left\{ \frac{1}{\psi} \frac{\partial^2 \psi}{\partial z^2} \right\} z \bar{z} \\
+ \frac{1}{4 \pi} \text{Im} \left\{ \left( \frac{1}{\psi^2} \frac{\partial \psi}{\partial z} \frac{\partial \psi}{\partial \bar{z}} \right) z \bar{z} \right\} &= 0 \\
\frac{1}{4 \pi} \text{Im} \left\{ \left( \frac{1}{\psi} \frac{\partial \psi}{\partial z} \right) \frac{1}{z} \right\} &= 0
\end{align*}
\]

We have studied exact integrals for four different flow geometries and Hamel's problem [1] in this work. The novelty of this work is in its approach since the Hamel's problem that has been investigated here was also investigated by Chandna and Toews [2]. The approaches used in the previously published works required the transformation of the flow equations to curvilinear coordinates when the streamlines and their orthogonal trajectories formed the coordinate net. No such transformation is required in the present complex variable approach.

The streamlines and their orthogonal trajectories form an isometric net for incompressible and irrotational steady plane flows. G. Hamel investigated those steady plane rotational fluid motions for which the streamlines and their orthogonal trajectories form an isometric net. This problem is called Hamel's problem. Martin [3] gave new formulation of the Navier-Stokes equations and studied Hamel's problem as an application of his approach. His method also required the transformation of the flow equations to the streamline curvilinear coordinates. Complex variable technique employed in this paper is well known for the analysis of fluid dynamic problem. Wan-Lee Yin [4], Ratip Berker [5] and Stallybrass [6] have employed this technique in their recent researches.

References

ABSTRACT - This paper presents an analysis for the superharmonics of a forced nonlinear vibration problem involving small parameters, using a generalized harmonic balance method. A nonlinear ordinary differential equation with several nonlinear terms and a periodic forcing function is considered. For the case of superharmonic oscillations of order 2, the key equations for the obtaining the information on the superharmonics will be derived, including a new, nonlinear ordinary differential equation of a slow varying function compared with the original dependent variable. Using these equations, the steady state solution and its stability behavior can be calculated. Results for a special set of parameters are obtained, including a stable node for the steady state solution and the associated van der Pol plane.

1. INTRODUCTION

It is well known that nonlinearities can cause sub- and super-harmonic excitations in vibratory systems. The analytical understanding of such phenomena is often difficult to obtain. It has been shown that the method of multiple scales can be used to solve such problems as demonstrated in several papers by Nayfeh [1,2]. However, the procedures involved are quite complicated and requires recursive solution of differential equations, the elimination of secular terms and reconstitution, all of which are nontrivial procedures. More recently, in a paper by Noble and Hussain [3], an expansion method was introduced together with suggestions of several other approaches which may be used as alternatives to obtain pertinent information. One of these is the generalized harmonic balance method (GHB) [4,5,6]. This variant of the harmonic balance method consists of two parts: first, to derive the form of solution using only the basic steps of multiple scales, and then, solve for the coefficients of various harmonics. In this approach, the elimination of the secular terms is accomplished implicitly, thus avoiding the trouble of solving recursive differential equations. This paper begins with a general nonlinear ordinary differential equation with several nonlinear terms and a periodic forcing function, a specific case of superharmonic oscillations of order 2 will be investigated. Next, the key equations are derived, from them the essential information on the superharmonics can be obtained. Finally, numerical results are presented on the steady solution and the stability behavior for a special set of parameters.

2. DERIVATION OF THE KEY EQUATIONS

We shall consider the following rather general differential equation:

\[ \frac{d^2u}{dt^2} + u + 2\omega_0(\frac{du}{dt}) + \omega_0^2u + \epsilon^2a_ku^3 + \epsilon(\frac{d^2u}{dt^2} + 2i\epsilon\omega_0u + \epsilon\omega_0^2u(\frac{du}{dt})^2) A + cc \]  

where \( u(t) \) is the unknown function, \( \epsilon \) and \( a_k \), \( k=2,3,4,5 \) and 6, are given constants, \( \epsilon \) is the small perturbation parameter; \( f \) and \( \omega \) pertain to the magnitude and frequency of the forcing function. For superharmonics of order 2, one has

\[ 2\omega = \omega_0 + \epsilon \sigma = 1 + \epsilon \sigma \]  

where \( \omega \) is the "fundamental" frequency of the nonlinear vibration, which is a perturbation from that of the linearized system \( \omega_0 \), taking to be unity in (2) without a loss of generality.

We shall derive a two-term approximate solution \( u=U_0+\epsilon u \), for equation (1). Using a procedure described previously in [4,5], it can be shown easily that that the final form of the solution \( u \), which is good to the order of \( \epsilon \) must have the following form:

\[ u=U_0 + [(U_0 + U_0^2 + U_0^4 + \epsilon U_0^3 + \epsilon U_0^5 + \epsilon^2 U_0^7) A + cc] \]  

where \( cc \) stands for the complex conjugate. The following symbols have been introduced:

\[ A=\exp(\epsilon t/2), \quad S=\exp(\epsilon \omega t/2) \]

Eq. (1) can then be written as

\[ \frac{d^2u}{dt^2} + U_0^2 + 2\epsilon U_0(\frac{du}{dt}) + \epsilon^2u^2 + \epsilon^2U_0^2(\frac{du}{dt})^2 + fA^2 + cc \]  

Here we note that \( S \) is a slow varying function compared with \( A \) in the sense that while \( dA/dt \) is of \( O(1) \), \( dS/dt \) is of \( O(\epsilon) \). We shall also use the fact that

\[ \overline{A} = e^{-i\omega t/2}, \quad A \overline{A} = 1 \]

where an overbar denotes the complex conjugate. The procedure here is to substitute (3) in (1') and set to zero the coefficients of \( A^k \), \( k=0, 1 \) and 2, since any higher harmonics will be of \( O(\epsilon^3) \) or higher according (4). We first obtain the following approximate expressions (in other words, the right hand side should have added \( + \) terms of \( O(\epsilon^3) \) and higher) in each of these equations:

\[ \frac{du}{dt} = (du_0/dt + iU_0)A + \epsilon(c_0u_0/dt + (d^2u_0/dt^2 + 4idu_0/dt - U_0) A) \]  

\[ \frac{d^2u}{dt^2} = ed^2U_0^2/dt + (d^2u_1/dt^2 + 2idU_1/\omega dt - U_1)A + \epsilon(d^2U_0^2/dt^2 + 4idU_0/dt - 4U_0^2)A^2 + cc \]  

\[ u^2 + 2U_0U_1 + U_1^2A^2 + 2c(U_0U_1 + U_0U_1 A) + cc \]  

Since \( u^3 \) appears with a coefficient of \( \epsilon^2 \) in (1), one only needs to keep terms of \( O(1) \) in the expansion:

\[ u^3 = 3U_0^2U_1 + A + cc \]

Similarly, one keeps \( O(\epsilon) \) terms in \( (du/dt)_2 \), but only \( O(1) \) terms in \( u(du/dt)_2 \):

\[ (du/dt)_2 = 2U_0U_1 - (U_0^2A + cc) \]

\[ u(du/dt)_2 = U_1U_1A + cc \]
We now substitute (4) and (6)-(11) in (1'), collect terms of like power of $A^k$, $k=0,1$ and 2, and then set the coefficients to zero. The resulting equations, for the coefficients of $A^{0}$, $A^1$ and $A^2$, respectively, are:

\[ e(U_0 + 2(\alpha_2 + \alpha_4)U_1 U_2 + (1/2)(4\alpha_2 + \alpha_4)U_2 U_2) = 0 \]  
\[ 3U_1 /4 - fS + dU_1 /dt + \epsilon \mu U_1 + e(2\alpha_2 + \alpha_4)U_1 U_2 = 0 \]  
\[ 21(dU_2 /dt + \epsilon \mu U_2) + e(4\alpha_2 - \alpha_4)U_2^2 /4 + d^2U_2 /dt^2 + \epsilon(2\mu U_2 /dt + \epsilon \alpha \mu U_1 U_1 /dt \]  
\[ + e^2(2\alpha_2 U_2 + (2\alpha_2 + 3\alpha_4 /2)U_1 U_2 + 2(\alpha_2 + 2\alpha_4)U_1 U_4 + \{3\alpha_2 + \alpha_4\}U_2^2 U_2 + \{5\alpha_2 + \alpha_4 /2\}U_1 U_2^2) = 0 \]  
\[ -5\epsilon U_4 /4 + e(2\alpha_2 - \alpha_4)U_1 U_2 = 0 \]  
\[ -3\epsilon U_4 + e(\alpha_2 - \alpha_4)U_2^2 = 0 \]  
\[ (14) \]
\[ (15) \]
\[ From (12), (15) and (16), U_0, U_1 and U_2 can be solved directly in terms of U_1 and U_2: \]
\[ U_0 = -2(\alpha_2 + \alpha_4)U_1 U_2 - (1/2)(4\alpha_2 + \alpha_4)U_2 U_2 \]  
\[ U_1 = (4/5)(2\alpha_2 - \alpha_4)U_1 U_2 \]  
\[ U_2 = (\alpha_2 - \alpha_4)U_2^2 /3 \]  
\[ (17) \]
\[ (18) \]
\[ (19) \]

In equation (13) and (14), however, it is observed that some terms are of one order of $\epsilon$ greater than the others. The terms of higher order in $\epsilon$ can thus be less accurate than others and still yield the same order of approximation in these equations. One then can solve these equation first using only the dominant terms. Then, substitute the results back into the terms of higher order in $\epsilon$, solve the full equations and obtain improved results. The immediate purpose here is to reduce (16) into a first order differential equation in $U_1$ and express all the other $U_k$s in terms of $U_2$.

Using the dominant terms in (13) and (14), one has

\[ U_1 = 4fS/3 \]  
\[ 21(dU_2 /dt + \epsilon \mu U_2) + e(4\alpha_2 - \alpha_4)U_2^2 /4 = 0 \]  
\[ (20) \]
\[ (21) \]

Equation (20) is used in the terms of order $\epsilon$ in (13) to yield the improved $U_1$:

\[ U_1 = 4fS/3 + (1/9)\epsilon [8(\sigma - 2i\mu)fS \]  
\[ - 16(2\alpha_2 + \alpha_4)fS\mu U_2 = 0 \]  
\[ (22) \]

Now, the terms in (16), which are of higher order in $\epsilon$, contain such quantities as $d^2U_2 /dt^2$, $dU_2 /dt$, $dU_1 /dt$, $U_1$, $U_1$, $U_2$. These expressions can be obtained by using (21), (22), their differentiations (for $d^2U_2 /dt^2$ and $dU_2 /dt$), (17), (18) and (19). The final form of (16) can be written as the following:

\[ 2i\xi U_2 /dt + \epsilon(2i\mu U_1 + c_1 \xi^2 S^2 U_2) \]  
\[ + e^2[c_2 U_2^2 U_2 + c_3 \xi^2 S^2 U_2 + (c_4 \xi^2 - \mu^2)U_2] = 0 \]  
\[ (23) \]

where

\[ c_1 = 4(4\alpha_2 - \alpha_4) /9 \]  
\[ c_2 = (9\alpha_2 + 3\alpha_4 - 10\alpha_2^2 - 10\alpha_4 \alpha_4 - 4\alpha_4^2) /3 \]  
\[ c_3 = c_3 + i\xi \]  
\[ (24) \]
\[ with \]  
\[ c_3 = 2(20\alpha_2 - 17\alpha_4) /27 \]  
\[ c_4 = -2\mu(52\alpha_2 - 13\alpha_4) /27 \]  
\[ and \]  
\[ c_5 = (1440\alpha_2 + 120\alpha_4 - 1472\alpha_4) /368 \alpha_2 - 128\alpha_4^2 /135 \]  

The key equations (4), (16), (17), (18), (19) and (22) can be further simplified by the following change of variables. Let

\[ U_k = V_k e^{\xi t}, \quad V_k = u_k - e^{\xi t}, \quad k=0,1,2 \]  
\[ (25) \]

where $S$ has been defined in (6). One also has

\[ dU_k /dt = dV_k /dt + i\xi \sigma V_k /2 \]  
\[ (26) \]

In terms of $V_k$, equations (4), (16), (17), (18), (19) and (22) become respectively

\[ u = \epsilon V_0 + \epsilon V_1 + \epsilon V_2 B^2 + (c_3 B^3 + c_4 B^4) + c_5 \]  
\[ (27) \]

with

\[ V_0 = - (32/9) (\alpha_2 + \alpha_4) \xi^2 - 2(\alpha_2 + \alpha_4) V_2 \]  
\[ V_1 = 4fS/3 + (1/9) \epsilon [8(\sigma - 2i\mu)fS - 16(2\alpha_2 + \alpha_4) \epsilon V_2 = 0 \]  
\[ (29) \]
\[ V_2 = (4/5)(2\alpha_2 - \alpha_4) V_1 V_2 \]  
\[ (30) \]
\[ V_4 = (c_2 - \alpha_4) V_2^2 /3 \]  
\[ (31) \]

and

\[ 2iV_2 /dt + 2(\sigma - 2i\mu + c_2 \xi^2) V_2 \]  
\[ + e^2[c_2 V_2^2 U_2 + c_3 \xi^2 U_2 + (c_4 \xi^2 - \mu^2)U_2] = 0 \]  
\[ (32) \]

where, in (29),

\[ B = SA = \exp[(1+i\sigma /2)t] = e^{i\sigma t} \]  
\[ (33) \]

Hence the original differential equation (1) has been reduced to (32), where $V_k$ is the unknown function. Once $V_k$ is solved, other $V_k$s can be obtained from (20) through (31). Then $u(t)$ is given by (27).

To illustrate what kind of information one can extract from the equations derived so far, we shall obtain the magnitude for a superharmonic in the steady state solution and determine the stability of such a solution. First, we shall write the needed equations in terms of real variables. To this end, let

\[ V_2 = V_{2R} + iV_{2I} = p_2 \exp(i\gamma V_2) \]  
\[ (34) \]
\[ V_2 = (x - iy) /2 \]  
\[ (35) \]

where now $p_2$, $V_{2R}$, $x /2$ and $V_{2I} = y /2$ are all real functions of t. One also has

\[ dV_2 /dt = (dx /dt - idy /dt) /2 \]  
\[ (36) \]

Note that we have introduced two new variables $x$ and $y$ such that

\[ x = 2V_{2R}, \quad y = 2V_{2I} \]  
\[ (36) \]
to save some writing. Substitute (34) and (35) in (32) and separate the real and imaginary part, one has two equations for two real variables x and y:

\[
\begin{align*}
\frac{dx}{dt} &= \epsilon \mu x + \epsilon^2 c_4 \ell^2 - c_2 (x^2 + y^2) y/8 + (c_5 \ell^2 - \mu^2) y/2 = 0 \\
\frac{dy}{dt} &= \epsilon \nu y + \epsilon^2 c_4 \ell^2 + \epsilon (c_5 \ell^2 + c_2 (x^2 + y^2)) x/8 + (c_5 \ell^2 - \mu^2) x/2 = 0
\end{align*}
\] (37a)

For steady state solutions, we require that the amplitudes and phase angles of various harmonic components to be constant with respect to time t,

\[
\frac{d\phi_k}{dt} = 0, \quad \frac{d\nu_k}{dt} = 0, \quad k = 0, 1, \ldots, 4
\] (38)

In particular,

\[
\frac{d\phi_2}{dt} = 0, \quad \frac{d\nu_2}{dt} = 0
\] (39a)

and, what is equivalent:

\[
\frac{dx}{dt} = 0, \quad \frac{dy}{dt} = 0
\] (39b)

It should be noted that (39a) actually also guarantee the validity of (38) for k other than 2. This fact can be easily observed from the relations of (28)-(31), which relate \( V_k \), \( k = 0, 1, 3 \) and 4, to \( V_2 \).

Now, substitute (39b) in (37), one has

\[
\begin{align*}
\mu x + \nu y &+ \epsilon (c_4 \ell^2 - c_2 (x^2 + y^2)) y/8 \\
+ (c_5 \ell^2 - \mu^2) y/2 = 0 \\
\mu y - \sigma x + c_4 \ell^2 + \epsilon (c_5 \ell^2 + c_2 (x^2 + y^2)) x/8 \\
+ (c_5 \ell^2 - \mu^2) x/2 = 0
\end{align*}
\] (40a)

Some numerical results will be presented in determining the presence of superharmonic oscillations for the following given set of parameters:

\[
\alpha_2 = 0.3, \quad \alpha_3 = 0.1, \quad \alpha_4 = 0.0, \quad \alpha_5 = 0.0, \quad \mu = 2.0, \quad \sigma = 3.0, \quad \ell = 2.0
\] (41)

This is a very simple case due to the fact that \( c \) vanishes as can be seen from (24). Thus (40) become linear and the solution can be easily obtained as

\[
x = 0.1824, \quad y = -0.0418
\] (42)

Hence, from (33), the magnitude of the superharmonic oscillation of order 2, \( \rho_2 \) is

\[
\rho_2 = 0.5 (x^2 + y^2) = 0.3754
\] (43)

Next, equations (37) are integrated numerically. The result is the so called van del Pol plane [7] as show in Figure 1. As indicated in this plot, solutions converge to the steady state solution obtained above as the time increases. Hence the steady state solution is stable and the point "A" of (42) is known as a stable node.

Results for more general cases will be reported in the future.

\[ FIGURE 1. \text{The van del Pol plane for the superharmonics of order 2 for the set of parameters given in Eqn. (41). Point "A" shown is a stable node.} \]

REFERENCES


DIFFRACTION ON A PERIODIC SURFACE

Andrew G. Mikheev and Aleksey S. Shamaev
Institute for Problems in Mechanics
USSR Academy of Sciences
Pr. Vernadskogo 101, Moscow 117526, USSR

Abstract In this paper the two-dimensional problem of diffraction of a plane electromagnetic wave on a smooth 2π-periodic surface is considered. The numerical method solving the problem of diffraction is developed.

1. Mathematical formulation of the problem.

The unknown function $u$ satisfies the Helmholtz equation

$$\Delta u + k^2 u = 0$$

in the region $\Omega = \{x, y\mid -\infty < x < f(y), 0 \leq y \leq 2\pi\}$. Here $k$ is wavenumber, $k = \frac{\omega}{c}$, $f(y)$ is smooth 2π-periodic function. The boundary condition for the function $u$ is:

$$\frac{\partial u}{\partial n} - h \cdot u(f(y), y) = 0$$

In the region $x < x_0 = \inf f(y)$ the radiation condition

$$u = e^{iK(x \cdot \cos \alpha + y \cdot \sin \alpha)} +$$

$$+ \sum_{n=-\infty}^{+\infty} T_n e^{-i\lambda_n y} x^{+\alpha \cdot y}$$

is imposed on $u$. Here $\alpha$ is the angle between the wave vector of incident wave and $x$-axis, $\lambda_n = k \cdot \sin \alpha + n, \gamma_n = \sqrt{k^2 - \lambda_n^2}, \Re \gamma_n \geq 0, \Im \gamma_n \geq 0$. $T_n$ are unknown amplitudes of scattered plane waves. The function $u$ is also assumed to satisfy the Focke conditions:

$$u(x, 2\pi) = u(x, 0) \cdot e^{i\eta t}$$

(4)

$$\frac{\partial u}{\partial y}(x, 2\pi) = \frac{\partial u}{\partial y}(x, 0) \cdot e^{i\eta t}$$

(5)

where $t = 2\pi k \cdot \sin \alpha$.


With the help of Green's function of Flocke canal

$$G(M, P) = \frac{1}{2\pi} \cdot \sum_{m=-\infty}^{+\infty} \frac{i\lambda_m \delta y \cdot i\gamma_m |\delta x|}{\gamma_m}$$

(6)

( here $\delta x = x_M - x_P, \delta y = y_M - y_P$ )

the problem is reduced to the one-dimensional integral equation for the $u(f(y), y)$:

$$\frac{1}{2} \cdot u(f(Y_M), Y_M) - \frac{1}{2\pi} \cdot \int_{0}^{2\pi} K\left[Y_M, Y_P, I(Y_P)\right] \cdot u(f(Y_P), Y_P) dY_P = e^{-1}$$

(7)

Here $I(y) = \sqrt{1 + [f'(y)]^2}$.

$$K\left[Y_M, Y_P\right] = h \cdot G(f(y_M), Y_M \cdot f(y_P), Y_P) -$$

$$- \frac{\partial g}{\partial y}(f(y_M), Y_M \cdot f(y_P), Y_P)$$

The integral equation (7) was solved with the help of method-of-moments:

Following [1], let us divide the segment $[0, 2\pi]$ into $N$ equal length segments, using points $y_i$ ($y_0 = 0, y_N = 2\pi$). Consider functions:

$$\phi_i(y) = \left\{\begin{array}{ll}
1, & y \in [y_{i-1}, y_i] \\
0, & y \in [y_{i+1}, y_i]
\end{array}\right.$$ (8)

Let us seek an approximate solution of equation (7) in the following form:

$$\psi^N(y) = \sum_{i=1}^{N} D^N_i \cdot \phi_i(y)$$

where coefficients $D^N_i$ are to be determined.

Function $\psi^N(y)$ is assumed to satisfy equation (7) in points $y = \frac{1}{2}(y_{i-1} + y_i)$. This gives algebraic equations for $D^N_i$ coefficients determination.

The expressions (9)-(14) give the well convergent series, which gives us the method for calculating the kernel of the integral equation (7).

$$G(M, P) = \frac{i\lambda \delta y \cdot i\gamma |\delta x|}{2\pi} +$$

$$+ \sum_{n=1}^{\infty} \left[ e^{ib\delta y} \cdot \left\{ \begin{array}{ll}
\frac{i\delta y}{\delta x} \cdot \xi_1(n, \delta x, \delta y)
\end{array} \right. -
\frac{i \cdot \sin(2\delta y)}{2} \cdot \xi_1(n, \delta x, \delta y)
\right] +$$

$$R(n, \delta x, \delta y)$$

(9)

Here $M = (x_M, y_M)$, $P = (x_P, y_P)$, $b=2\pi$.

$$\xi_1(n, \delta x, \delta y) = \cos(n\delta y) \cdot e^{-n|\delta x|}$$

(10)
\[ \bar{\varepsilon}_2(n, \delta x, \delta y) = \sin(n\delta y) e^{-n|\delta x|} \]  

(11)

Expressions (10), (11) can be summed:

\[ \sum_{n=1}^{\infty} \bar{\varepsilon}_1(n, \delta x, \delta y) = -\ln2 + \frac{\delta x}{2} \quad - \frac{1}{2} \ln \left[ \sin^2 \left( \frac{\delta x}{2} \right) + \sin^2 \left( \frac{\delta y}{2} \right) \right] \]  

(12)

\[ \sum_{n=1}^{\infty} \bar{\varepsilon}_2(n, \delta x, \delta y) = \quad \]  

(13)

\[ \frac{\partial G}{\partial n_p} = \frac{\partial}{\partial n_p} \left[ \frac{ie \delta y \delta y}{2\gamma_o} \right] + \]  

\[ \sum_{m=1}^{\infty} \left\{ e^{\frac{i\pi \delta y}{\delta x}} \left[ \frac{\partial \bar{\phi}_2(m)}{\partial n_p} - i \cdot \text{sh}(\frac{\delta x}{2}) \cdot \frac{\bar{\phi}_2(m)}{\partial n_p} \right] \right\} = \]  

\[ \text{sign}(\delta x) \cdot \text{ch}(\frac{\delta x}{2}) \cdot n_x + i \cdot \text{sh}(\frac{\delta x}{2}) \cdot n_y + \]  

\[ + \bar{\phi}_2(m) \cdot \left( \text{ch}(\frac{\delta x}{2}) \cdot n_y + i \cdot \text{sh}(\frac{\delta x}{2}) \cdot \text{sign}(\delta x) \cdot n_x \right) \]  

(14)

Here \( R \), \( Q \) satisfy the expressions (15):

\[ |R(n, \delta x, \delta y)| < \frac{C}{n}; |Q(n, M, P)| < \frac{C}{m^2} \]  

(15)

The kernel of the integral equation contains logarithmic singularity, which is expressed in explicit form in (12).


Figures 1, 2 show a peculiar example of distribution of electromagnetic surface current (function \( u(f(y), y) \)).

Here \( \kappa = \frac{3}{2}, \alpha = 60^\circ, h = (1+i) \cdot 0.075, f(y) = \frac{1}{2} \cdot \sin y \).

Parameter \( N = 30 \). Energy error equals \( 2 \cdot 10^{-4} \). Share of reflecting energy \( = 0.28 \).

Figures 3, 4 show a typical example of distribution of electromagnetic surface current.

Here \( \kappa = \frac{3}{2}, \alpha = 75^\circ, h = 0 \).

\[ f(y) = \frac{1}{2} \cdot \sin y \].

Parameter \( N = 30 \) (see (8)).

Energy error equals \( 4 \cdot 10^{-4} \).

REFERENCES

Simulation of Distributed Feedback Dye Laser by Computer

by J. Seres

Department of Physics, Juhász Gyula College
Boldogasszony sgt. 6. PO Box 366 Szeged, H-6701 Hungary

Abstract - A numerical method has been developed for computing coupled partial differential equations describing distributed feedback dye lasers. This method is founded on the Euler method, but is faster about 30 times. The necessary computation on the time and space axes has been determined for the required accuracy computations. Calculations have been made by the novel method and the obtained results have been compared with earlier published measurements. The computed results have shown good agreements with measured values.

I. INTRODUCTION

The arrangements and behaviours of DFDLs have been extensively studied since 1971. A typical arrangement of the DFDL and several tuning possibilities are shown by [3]. The processes happening in DFDLs may be described by coupled differential equations (rate equations) [1,3,5,8,9]. For the most part these equations contain physical quantities depending only on the time variable. Many calculations were made and their results were compared with measured values in last years. The experiences show if the pulse duration of the exciting laser and DFDL is longer many times than the time calculated by the laser length and the refractive index then computed values approximate the values of measurements [1,2,3]. If these conditions don't exist the measured values are fully different from calculated ones. In this case the calculations give exact values if the physical quantities describing DFDL depend on space variable too [9]. For two variables the running time is many times longer than it for one variable. A novel algorithm has been developed to reduce the running time. This algorithm based on the Euler method, but faster than it many times.

II. MODEL OF DFDL

Coupled partial differential equations are shown below [1,2]. These equations noted down with the help of published equations in [1,3,5,8,9] describe the processes in DFDLs.

\[
\frac{\partial N(x,t)}{\partial t} = I(x,t) - \sigma_p \left( N_0 - N(x,t) \right) - \frac{N(x,t)}{\tau} - \sigma_a N(x,t) \left( I(x,t) + I^e(x,t) \right) \quad (1)
\]

\[
\frac{\partial I(x,t)}{\partial x} + n \frac{\partial I(x,t)}{\partial t} = \Omega N(x,t) + \sigma_a \tau \left( I^e(x,t) - I^s(x,t) \right) \quad (2)
\]

The meaning of the symbols are as follows:

- \( N_0 \): the density of dye molecules \( [2.1 \cdot 10^{24} \text{m}^{-3}] \),
- \( N(x,t) \): the density of molecules in the \( S_1 \) excited state \( [\text{molecules}\cdot\text{m}^{-3}] \),
- \( I(x,t) \): the density of the DFDL photon current propagating into the +x and -x direction, respectively \( [\text{photons}\cdot\text{m}^{-2}\cdot\text{s}^{-1}] \),
- \( I_p(t) \): the density of the pump photon current \( [\text{photons}\cdot\text{m}^{-2}\cdot\text{s}^{-1}] \),
- \( \sigma_p \): the absorption cross section from \( S_0 \) at the pumping wavelength \( [2.4 \cdot 10^{-24} \text{m}^2] \),
- \( \sigma_e \): the stimulated emission cross section from \( S_1 \) to \( S_0 \) state at the lasing wavelength \( [1.4 \cdot 10^{-20} \text{m}^2] \),
- \( \sigma_a \): the excited state absorption cross section from \( S_1 \) to \( S_2 \) at the lasing wavelength \( [0.7 \cdot 10^{-20} \text{m}^2] \),
- \( \tau \): the fluorescence lifetime of \( S_1 \) state \( [4 \text{ns}] \),
- \( n \): the refractive index of the dye solution \( [1.44] \),
- \( c \): the speed of light in vacuum,
- \( V \): the visibility of the amplitude-phase grating in the excited volume \( [1] \),
- \( \Omega \): the factor determining that fraction of the spontaneous emission which propagates into the angular and spectral range of the DFDL beam.

The value of \( \Omega \) is calculated as \( \Omega = \frac{b \cdot a}{\pi \cdot L^2 \cdot S} \), where

- \( L \) is the length of DFDL,
- \( a = (N_0 \cdot \sigma_p)^{-1} \) is the penetration depth of the pumping beam into the dye solution,
- \( b \) is the height of the excited volume, and
- \( S \) is the spectral factor determining that.
fraction of the spontaneous emission, which falls into the DFDL bandwidth [1,3]. This parameters described the dye laser what contained the Rhodamine 6G dissolved in methanol and what was excited by $N_2$ laser. The wavelength of $N_2$ laser was 337.1 nm, and the pulse duration was 3.5 ns.

III. LIMIT OF $\Delta x$ AND $\Delta t$

Calculations have been made to determine the limits of the distributing intervals on both time and space axes. The rate equations depending only time variable have been used for determining the maximum of time intervals ($\Delta t$). The energy and the duration of the first DFDL pulse have been computed at the threshold pump intensity what is necessary to come out the second pulse. Computations have been made at several $\Delta t$, and the divergences from exact value (where $\Delta t \rightarrow 0$) have been determined. The results calculated with both explicit and implicit Euler method show it is pay to use the implicit method, and if $\Delta t \leq 3$ ps then both energy and duration of the output pulse approximate the exact value under 1%. For determining the maximum of the space intervals ($\Delta x$) a pulse was propagated along the excited volume. The rate of output and input pulse intensity was calculated with both numerical and analytical method. The results calculated with numerical method have been under 1% if $\Delta x \leq L/550$. According to the limit of the time intervals it is necessary to distribute the length of laser ($L = 5.5$ mm) into 9, and according to the limit of the space intervals into 550.

IV. NOVEL ALGORITHM

Let the length of laser is distributed into $k$ intervals and every intervals are distributed into $m$. The $k$ is determined by limit of time intervals and the $m$ by limit of space intervals. The algorithm has been written down after longer considerations:

\[
I_{i+m,t}^+ = \frac{1}{\nu} \left[ \frac{\sigma - \sigma_0}{\alpha} \Delta x \right] \cdot I_{i,t}^+, \quad \sigma = \frac{\nu}{m} I_{i,t}^+, \quad \nu = \frac{\nu^2}{\sigma_0}
\]

where $\Delta x = 1 + (\sigma - \sigma_0$) $\Delta t$, $I_{i,t}^+$ is the intensity of the pulse at the time $t$ and $\sigma_0$ is the interaction parameter.

\[
\text{The $\xi^n$ is quickly calculated if the $m = 2^J$, where $J$ is an integer number.}
\]

![Graph of measured and calculated pulse duration of the single pulses from a Rhodamine 6G DFDL.](image)

Fig. Measured and calculated pulse duration of the single pulses from a Rhodamine 6G DFDL. The pump intensity was adjusted to the threshold of the second DFDL pulse.

V. RESULTS OF CALCULATIONS

The $k = 10$ and $m = 64$ has been chosen for the computations. The measured and the calculated values are shown on the figure. According to the figure the calculated values are in good agreement with the measurements [1].

References:
4. J. Hebling: Optics Comm. 64, 539-543 (1987)
ADAPTED VERSIONS OF THE EN ALGORITHM FOR PENALIZED LIKELIHOOD IN EMISSION TOMOGRAPHY

Alvaro R. De Pierro
Institute of Mathematics, Statistics and Computer Science
State University of Campinas
CP 6065, 13081, Campinas, SP, Brazil

Abstract. We present in this paper two different methods for solving the penalized likelihood maximization problem arising in emission computed tomography (ECT). Both methods are modifications of the Expectation Maximization (EM) algorithm proposed to overcome the inability of this algorithm in its usual form to cope with penalization terms in a non expensive way.

I. INTRODUCTION

In ECT we aim to reconstruct a function that is the distribution of radioactivity in a body cross-section and the measurements are used to estimate the total activity along lines of known location. Higher levels of noise induce the use of mathematical models incorporating the statistical nature of the process instead of inverting the Radon transform as in x-ray computed tomography [1]. In [2], Shepp and Vardi, suggested the use of maximum log-likelihood estimates derived from the Poisson nature of the emission process, i.e.,

\[
\max_{x \in \Omega} L(x) = \sum_{i=1}^{n} y_i \ln a_i(a_i \cdot x) - <a_i, x>,
\]

where \( y = \{y_i\} \) (1 = 1, ..., m) are the photon counts, \( a_i \) are the columns of \( A = \{a_{ij}\} \), the matrix modelling emission features, and \( x = \{x_j\} \) (j = 1, ..., n) the image vector (emission density) to be reconstructed \(<,> \) denotes the standard inner product and \( \sum_{j=1}^{n} a_{ij} = 1 \).

To solve (1), Shepp and Vardi proposed to use the EM algorithm obtaining very good results for earlier steps. Unfortunately, iterations have to be stopped before a deteriorating effect (irregular high amplitude patterns) appears. To cope with this inconvenient, quadratic penalization terms have been suggested [3], so problem (1) turns now to be

\[
\max_{x \in \Omega} L(x) = \frac{1}{2} p(x),
\]

where \( p(x) \) is a convex quadratic function and \( \gamma \) a positive parameter. For the new problem, the standard EM algorithm is no longer applicable in practice except if the matrix associated with \( p(x) \) is diagonal. In the following sections we describe the EM algorithm for (2) and for the pure quadratic problem, as well as the new alternatives proposed for (2).

II. THE EM ALGORITHM

Let \( Y \) be a random vector (observed data in some experiment) with density function \( g(Y,x) \), where \( x \) is some vector of parameters to be estimated. If \( g \) is difficult to maximize with respect to \( x \), a possible solution is to embed \( Y \) in a richer sample space \( X \) where the optimization problem is easier to solve. Then, the EM algorithm is defined as, given \( x \in \Omega \), no parameter space

\[
x^{k+1} = \arg \max_{x \in \Omega} q(x|x^k), \text{ for } k=0,1,2,\ldots (3)
\]

where \( q(x|x^k) \) is the expectation of the extended log-density given \( (Y,X^k) \). Convergence properties and more details for (3) can be found in [4] and [5].
IV THE EXTENDED EM ALGORITHM

In spite of the advantage of being convergent for every \( \gamma \) the EPH algorithm is quite expensive because of the up-dating of the scalar products (see [7]). So we propose an alternative based on introducing a set of artificial data as described next.

\[ S = \frac{d}{\gamma} H, \text{ for some } H \in \mathbb{R}_{nxn}; \text{ on the other hand if } p \text{ has a minimum (standard assumption), } Sx = q \text{ has a solution and } q \in \mathbb{R}(H), \text{ i.e., } q = Hb \text{ for some } b. \]

Formally we can think that \( b \) is a normally distributed random vector and the same procedure as for (5) is applicable. Applying the expectation, first with respect to \( y \), afterwards with respect to \( b \), gives (combining (4) and (6))

\[
q(x/x^n) = \sum_{j=1}^{n} \sum_{i=1}^{m} \frac{a_{ij}y}{a_{ij} - \text{ln} a_{ij} x_j - a_{ij} x_j} - \gamma \sum_{i,j} (h_{ij} - h_{ij} x_j)^2
\]

\[ n \gamma s_{jj} x_j = 0. \] (11)

Differentiating, \( x_j^{k+1} \) will be the unique positive solution of (for \( j = 1, \ldots, n \))

\[
\frac{x_j^{k+1}}{x_j} = \sum_{i=1}^{m} \frac{a_{ij}y}{a_{ij} - \text{ln} a_{ij} x_j - a_{ij} x_j} - 1 - n s_{ij} x_j \gamma q_j - s_j x_j \]

\[
\frac{n}{\gamma} s_{jj} x_j = 0. \] (11)

Taking into account that \( s_{jj} = \sum_{i=1}^{m} h_{ij} \) and \( q_j = -s_j x_j \)

\[
= \sum_{i=1}^{m} h_{ij} (s_{ij} - h_{ij} x_j).
\]

Now variables are separated and (11) is a single unknown quadratic equation for each \( j \). Convergence of the algorithm is guaranteed because it is a special case of the EM algorithm.

Further work has to be done in order to test practical performance of this algorithm as well as extensions to other nonquadratic penalizations.

References.


BOUNDARY ELEMENT METHOD FOR NONLINEAR ELLIPTIC PROBLEMS WITH NONLINEAR MATERIAL CONDITIONS

K. ROOTSALEINEN
University of Oulu,
Faculty of Technology, Section of mathematics,
SF-90570, Oulu, Finland

Abstract. We shall consider the numerical analysis of the boundary element for nonlinear elliptic problems with nonlinear material conditions. The convergence of the Galerkin and collocation schemes is proved.

1. Introduction

In this paper we study the possibility to apply the boundary element methods to nonlinear elliptic boundary value problems with nonlinear differential equation. To this end in the analysis of the nonlinear boundary element method one has frequently restricted to the cases where the differential equations are linear. The nonlinearity appears only in the boundary conditions.

By means of the Kirchhoff transform we are able to linearize the differential equation. The introduced new unknown function satisfies a linear differential equation. The boundary conditions are, however, nonlinear. This can be done when the differential operator is in the divergence form and the nonlinearity depends only on the function itself; not on its derivatives.

By the indirect approach of the boundary integral equation methods the "linearized" problem can be transformed into a nonlinear boundary integral equation for the unknown boundary distribution which is to be solved numerically. We analyse both the Galerkin and collocation methods for finding an approximate solution. Here we present the convergence results and some preliminary error estimates. The complete analysis is presented in the forthcoming paper [4].

2. The Formulation of the Problem

We shall consider a nonlinear boundary value problem that is to be encountered in a stationary heat conduction problem with a temperature dependent heat conductivity. The problem consists in finding a potential function \( \phi \in W^{1,2}(\Omega) \) such that

\[
-\nabla \cdot (a(\phi) \nabla \phi) = 0, \quad \text{in } \Omega
\]

\[
-a(\phi) \frac{\partial \phi}{\partial n} = G(\phi)|_{\Gamma} - f, \quad \text{on } \Gamma.
\]  

(1)

Throughout the paper we assume that \( \Omega \) is a bounded plane domain with a regular boundary \( \Gamma \). In other words the boundary \( \Gamma \) has a regular parameter representation \( x : R \rightarrow \Gamma \) with a nonvanishing Jacobian; \( \frac{dx}{dt} \neq 0 \). The symbol \( \frac{dn}{dt} \) stands for the outer normal derivative as usual.

Before proceeding with the reformulation of the problem we recall that \( W^{m,p}(\Omega), 1 < p < \infty, \) is the usual Sobolev space with usual norm \( \|u\|_{W^{m,p}} \). Besides these spaces we need in the sequel the Sobolev-Slobodeckii spaces \( W^{m,p}(\Gamma) \) on the boundary. Other related function spaces are introduced in the order of occurrence.

Let us now consider the reformulation of the problem (1). For that we shall make the following basic assumption: We suppose that \( a(\phi) \) is a sufficiently smooth function and that there exists positive constants \( m \) and \( M \) such that for all \( \phi \in R \)

\[
0 < m \leq a(\phi) \leq M < \infty.
\]  

(2)

Now for the Kirchhoff transform defined by setting

\[
K(\phi) = \int_{\Gamma} a(\phi) d\sigma
\]

it holds [4]:

Lemma 1. Let the function space \( X \) be \( L^2(\Omega) \) or \( L^2(\Gamma) \), respectively. Then the Kirchhoff transform \( K : X \rightarrow X \) is Lipschitz-continuous and strongly monotone, i.e. for every \( u, v \in X \)

\[
(K(u) - K(v), u - v)_{\Gamma} \geq m \|u - v\|^2.
\]

Furthermore, the mapping \( K : W^{1,2}(\Omega) \rightarrow W^{1,2}(\Omega) \) is bijective.

By the theory of monotone operators we easily conclude that the inverse transform \( K^{-1} : L^2(\Gamma) \rightarrow L^2(\Gamma) \) is also a strongly monotone and Lipschitz-continuous mapping. With the previous lemma it is easy to verify that \( \psi = K(\phi) \) satisfies the following potential problem

\[
\Delta \psi = 0, \quad \text{in } \Omega.
\]

\[
\frac{\partial \psi}{\partial n} = G(K^{-1}(\psi)) - f, \quad \text{on } \Gamma.
\]  

(3)

provided \( \psi \) solves the problem (1). The converse also holds if \( \psi \) is the solution of our original problem (1). This is to say that problems (1) and (3) are equivalent in the weak form.

The problem (3) can be formulated as a nonlinear boundary integral equation [4]. This can be accomplished by introducing a boundary distribution \( u \) such that

\[
\psi(x) = -\frac{1}{2\pi} \int_{\Gamma} u(y) \log |x - y| \, ds_y, x \in \Omega.
\]

Then by the trace properties of the normal derivative of the single layer potential we derive the nonlinear boundary integral equation

\[
A(u) = (\frac{1}{2} I - D^*) u + G(K^{-1}(S(u))) = f.
\]  

(4)

where the operator \( D^* \) is the spatial adjoint of the double layer operator \( D \), which is defined by setting

\[
D u(x) = \frac{1}{2\pi} \int_{\Gamma} u(y) \frac{\partial}{\partial n} \log |x - y| \, ds_y,
\]

and \( S \) denotes the single layer operator defined as

\[
S u(x) = -\frac{1}{2\pi} \int_{\Gamma} u(y) \log |x - y| \, ds_y.
\]

For the unique solvability of the boundary integral equation (4) and of the boundary value problem, we make the following assumption

A1. The nonlinear operator \( G : L^p(\Gamma) \rightarrow L^q(\Gamma), 2 \leq p < \infty, \frac{1}{p} + \frac{1}{q} = 1, \) is bounded, continuous and strictly monotone. In addition to these we suppose that for almost all \( x \in \Gamma \)

\[
G(u, x) \geq a(u(x))^p + b, a > 0, b \in R.
\]
Then the boundary integral equation is uniquely solvable. More precisely we have [3],[4]:

**Theorem 2.** For every \( f \in L^q(\Gamma) \) there exists a unique solution \( u \in L^q(\Gamma) \) to (4).

3. **The Boundary Element Discretization**

As the approximation schemes we shall use the Galerkin and collocation methods. For the approximation we use the boundary element spaces \( S_h^d(\Theta) \) with respect to the partition \( \Theta = \{\xi_i = x(\tau_i) | i = 0, \ldots, N - 1 \} \) on \( \Gamma \), where \( \tau_i \) are grid points on the unit interval that is carried to \( \Gamma \) by the parameter representation. We assume that the family of the partitions is quasiuniform. We remind the reader that \( S_h^d(\Theta) \) corresponds via the parameter representation the smoothest 1-periodic splines of degree \( d \) on the unit interval [1]. As a mesh parameter we choose \( h = 1/N \).

The problem, in general, is to find the coefficients \( \alpha_i \in R, i = 0, \ldots, N - 1 \) such that \( u_h = \sum_{i=0}^{N-1} \alpha_i \psi_i \) is as good approximation of the true solution as possible. The functions \( \psi_i \) forms a suitable basis of the spline space. We shall present here the convergence analysis of the two most popular methods.

**The Galerkin method.** As it is well-known in the Galerkin method the coefficients are fixed by means of the orthogonality condition

\[
(A(u_h), \varphi)_{L^q(\Gamma)} = (A(u), \varphi)_{L^q(\Gamma)}, \varphi \in S_h^d(\Theta)
\]

This is the same as to treat the family of the operator equations

\[
\left( \frac{1}{2} I + P_h D^* \right) u_h + P_h G(K^{-1}(S(u_h))) = P_h f,
\]

where \( P_h : L^q(\Gamma) \rightarrow S_h^d(\Theta) \) is the orthogonal projection. In the convergence analysis we apply the theory of a-proper-mappings [2]. Since the nonlinear mapping \( A(\cdot) \) is an operator of Gårding type we have [4]:

**Theorem 3.** The nonlinear operator \( A(\cdot) : L^q(\Gamma) \rightarrow L^q(\Gamma) \) is a-proper with respect to the projectionally complete scheme \( \{P_h, S_h^d(\Gamma)\} \).

For the solvability of the Galerkin equations we shall need the following assumption, which usually valid in true applications.

**A2.** \( G : L^q(\Gamma) \rightarrow L^q(\Gamma) \) is continuously Fréchet-differentiable, and that the Fréchet-derivative \( DG(u) \) is strictly monotone (i.e. positive).

With this assumption the nonlinear operator has the following properties:

**Theorem 4.** The Fréchet-derivative \( DA(u) : L^q(\Gamma) \rightarrow L^q(\Gamma) \) is a Fredholm operator with vanishing index. \( \text{ind}(DA(u)) = 0 \) in addition to this the derivative is one-to-one.

As a corollary we conclude [4]:

**Theorem 5.** There exists \( h_0 > 0 \) such that for every \( 0 < h < h_0 \) the Galerkin method yields a unique solution to (5) ( or (6)) and \( \|u - u_h\| \rightarrow 0 \) as \( h \rightarrow 0 \).

Finally, utilizing the Fredholmness of nonlinear operator \( A(\cdot) \) and the approximation properties of spline spaces on \( \Gamma \) we are able to derive the asymptotic error estimates:

**Theorem 6.** Let \( u_h \) be as in the previous theorem. Then for sufficiently small \( h \) there holds

\[
\|u - u_h\|_{W^{s+1} \Gamma} \leq c(\|u\|_{L^q(\Gamma)}) h^{s-1} \|u\|_{W^{s+1} \Gamma}
\]

for every \(-1 \leq t \leq s \leq d + 1\).

**The collocation method.** For the collocation method we define the collocation points as follows:

\[
x_i = \frac{x(\tau_i) + x(\tau_{i+1})}{2}, d \text{ is even},
\]

\[
x_i = \frac{x(\tau_i) - x(\tau_{i+1})}{2}, d \text{ is odd},
\]

Here the points \( \tau_i \) on the real line corresponds the grid points of the partition \( \Theta \) via the parametrization of \( \Gamma \). If we let \( I_h \) to denote the interpolation operator, which interpolates between the function values on the location points \( x_i \), from the space of continuous functions \( u \rightarrow \) we can write the collocation equations in the form

\[
I_h A(u_h) = I_h f.
\]

The interpolation equation (7) is not always reasonable. Therefore we must have some additional properties besides the assumptions A1 and A2. We require that

**A3.** The mapping \( G : W^{1+\epsilon} \Gamma \rightarrow W^{1+\epsilon} \Gamma \) is bounded with some \( \epsilon > 0 \) that is sufficiently small \( (0 < \epsilon < \frac{1}{2}) \).

After this additional property the collocation equations (7) make sense. The more severe difficulty in the convergence analysis is due the fact that \( (I_h, S_h^d(\Theta)) \) is not a projectionally- complete scheme in \( L^q(\Gamma) \). However, since the collocation equations can be written as

\[
(I_h - I_h D^*) u_h + I_h G(K^{-1}(S(u_h))) = I_h f
\]

we don’t need the interpolation operator to be bounded in \( L^q(\Gamma) \). This is due the fact that the double layer and single layer operators are smooth operators. This allows us to extend the proof of the convergence of the Galerkin method to the collocation method. Thus we have [4]:

**Theorem 7.** There exists a mesh parameter \( h_0 \) such that for all \( 0 < h < h_0 \) the collocation equations admit a unique solution and \( \|u - u_h\|_{L^q(\Gamma)} \rightarrow 0 \) as \( h \rightarrow 0 \).

**References**


A Conceptual Architecture for Modelling Physical Systems

C. P. McGann and J. B. Grimson
Dept. of Computer Science,
Trinity College,
University of Dublin,
Dublin 2,
Ireland.
Tel: +353-1-772941
Fax: +353-1-772204
Email: CP-MCGANN@VAXI.TCD.IE

D.P. Finn
Hitachi Dublin Laboratory,
Hitachi Europe Ltd.,
O‘Reilly Institute,
Trinity College,
University of Dublin,
Dublin 2,
Ireland.
Tel: +353-1-6798911
Fax: +353-1-6798926
Email: DFINN@VAXI.TCD.IE

Abstract
A conceptual architecture for modelling abstract physical systems is described. The architecture aims to provide a system framework which is based on the engineering modelling process. This paper concentrates on the use of qualitative causal networks as a tool for predicting and estimating the physical phenomena acting within a physical system. Three qualitative simulation techniques are assessed and a constraint-based qualitative simulation approach is adopted. The incorporation of this approach within the overall system architecture is discussed.

1. Introduction

1.1 Background

Modelling of engineering problems can be divided into a number of stages: geometric modelling, physical modelling, mathematical modelling, numerical modelling and graphics modelling. Geometric modelling involves representing the geometric features of real world problems and if possible making geometric simplifications to reduce the complexity of the problem. Physical modelling involves identifying any physical phenomena which are occurring such as heat transfer, fluid flow or stress and if possible, making certain assumptions to simplify phenomena complexities. Mathematical modelling requires building suitable equations to represent the problem mathematically and selecting correct boundary conditions. Numerical modelling involves constructing suitable numerical algorithms and solving these algorithms computationally. Finally, graphical modelling requires the use graphical techniques to present the numerical solution. Many existing numerical problem solving environments such as DEQSOL [Umetani et al. 1985, Kono et al. 1986], ELLPACK [Rice 1985] and FIDISOL [Schonauer and S-Heap 1987], NEXUS [Gafney et al. 1986] help users with the mathematical and numerical modelling stages of the overall modelling process. Few systems, however, have focussed on the initial modelling stages, namely, geometric and physical modelling. In these modelling stages, the user is generally required to conceptually represent a real world problem, identify the nature and relative importance of the physical phenomena occurring and determine any geometric simplifications or phenomena assumptions which may be made to facilitate efficient numerical analysis.

1.2 Modelling of Physical Systems

This paper aims to address these shortcomings by presenting a conceptual architecture which addresses the process of engineering modelling for the problem domain described by partial differential equations. The discussion assumes as a starting point, a high-level, abstract, three dimensional representation of a real world problem or physical system (referred to as the complex model). This representation then evolves through a series of transition processes, each stage representing a simpler model. The final model (called the optimised model) is suitable for mathematical and numerical analysis. The architecture includes techniques for representing real world problems, identifying all physical phenomena and qualitatively simulating their behaviour. Additionally, the user may select significant phenomena for further analysis, carry out reduction of geometric and phenomena features to evolve an optimised geometry which is suitable for efficient mathematical and numerical modelling. Although all modelling stages are initially discussed in the paper, special emphasis is placed on the behavioural prediction of the physical phenomena. In particular, the incorporation of the two important artificial intelligence techniques of qualitative reasoning and causal network analysis are focused on to establish a high level, knowledge intensive network representation.

The paper is divided as follows; Section 2 gives an overview of the modelling process. Section 3 discusses other research which has addressed these higher-level stages of engineering modelling. The relevance of the proposed modelling methodology is illustrated and the incorporation of certain current research into the proposed architecture is discussed. Section 4 deals with qualitative reasoning as a key inference technique and discusses three principal qualitative reasoning methodologies. Section 5 discusses the fundamental approach taken in this work, i.e., qualitative causal networks. Section 6 concludes the paper.

2. System Outline

In this Section an overview of the important stages of the modelling process in the proposed architecture is presented. Furthermore, the intermediate stages of the modelling process are outlined. Having established a global perspective on the system architecture, system entities are introduced which form a framework for the modelling process.

2.1 System Architecture

Figure 1 gives an overview of the system architecture. Each stage depicts an evolution of the physical problem, beginning with the complex model at the highest level of abstraction and culminating in an optimised model which corresponds to the lowest representation level before mathematical modelling. Transition processes determine each evolution.

(i) Stage 1: 3-D Representation

The complex model is created by the user and is a representation of a real world problem or physical system as shown in Figure 2. The complex model consists of a number of components. Each component is represented as an object with initial attributes of geometry, function, location and material. At this stage the user is requested to specify any known phenomena features for each component. In the example illustrated in Figure 2, the user may be asked if an electric current, voltage difference or electric field are present.

(ii) Transition A: Cause-Effect Inference

A shallow rule based inference mechanism is used to exhaustively predict all possible phenomena in the system based on the high-level knowledge available from the 3-D representation.
Key
- 3-D
- Technique: Causal Reasoning
- Applied Phenomenon Prediction
- Transition: A

Cause-Effect Inference
- Phenomena Prediction
  - 3-D Representation Complex Model
  - Physical System Representation

Constraint based Qualitative Simulation. - Predicted Phenomena Primary Model
- Phenomena Quantification
  - Quantified Phenomena

High-level rule based inference - Phenomena Reduction
- Reduced Phenomena

Application Knowledge - Analysis Volume Reduction
- Optimised Model

Numerical Methods - Mathematical Modelling

Figure 1 Overview of proposed modelling architecture

(iii) Stage 2: Predicted Phenomena
At this stage, all known phenomena have been predicted by the system. However, it may not be realistic to analyse all phenomena, therefore by quantifying each phenomenon, certain phenomena may be found to be insignificant and therefore removed from further consideration.

Figure 2 Physical System or Real World Problem

(iv) Transition B: Constraint-based Qualitative Reasoning
Qualitative reasoning and causal networks are used to derive a behavioural description. Any simplifications such as geometric idealisations are carried out to derive a simplified primary model for each phenomenon; these simplifications are known as primary simplifications. Each phenomenon is then estimated using constraint-based qualitative reasoning to give a behavioural description. This description can then be assessed by the user to estimate the importance of each phenomenon. This approach is discussed in detail in Section 4.

(v) Stage 3: Quantified Phenomena
At this stage there is a stable description of the behaviour of all components of the simplified complex model.

(vi) Transition C: Phenomena Reduction
Shallow rule-based inference mechanisms are used in this transition to examine the values of key parameters for each phenomena throughout the system. Using knowledge bases, the relative importance of each phenomenon can be assessed, e.g., stress levels in electronic components and the user may make an informed selection of phenomena to include in subsequent analysis.

(vii) Stage 4: Reduced Phenomena
Stage 4 consists of a number of reduced phenomena for further analysis and optimisation.

(viii) Transition D: Phenomenon and Control Volume Reduction
Depending on the phenomena selected by the user for analysis, an optimised model is derived. Optimisation may consist of further geometric simplifications, selection of a subsection of the model based on symmetry or selection of a reduced analysis volume.

(ix) Stage 5: Optimised Model
This final stage is a model of the primary model optimised for subsequent analysis. It reflects a trade-off between simplicity and accuracy in analysis. For example, if the optimised model was to be used for numerical simulation, an important concern would be the scope of the analysis volume. If a large analysis volume is assumed, then the complexity of calculations is increased, whereas if the analysis volume is limited, important detail may be lost.

2.2 Introducing System Entities

The physical system is comprised of a number of components. Each component has a functional perspective and a geometric perspective. The functional perspective determines what use the component has. For example, a metallic pipe may be viewed as a flow channel for a fluid, an electrical conductor or a bearing for a rod that can rotate inside it. The geometric perspective defines the shape and dimensions of a component. Initially, a three-dimensional shape may be selected by the user from a limited library of shapes and instantiated with the appropriate dimensions e.g., a BLOCK of length A, width B and depth C. The geometric perspective evolves from the complex model through the primary simplification to the optimised model.

Each component has a behaviour. The behaviour of that component is essentially the physical phenomena occurring in that component and is context dependent. The context of a component determines the component’s behaviour and it is determined by the component’s geometric perspective, functional perspective and boundary links as well as its current local behaviour.

Boundary links are the medium through which effects are propagated between two connected components. A valid boundary exists between any two physically connected components and also between a component and the outside world, where they are directly in contact. A boundary link is an abstract entity which models the transfer of effects of physical phenomena between neighbouring components. Transfer parameters are defined for effects of physical phenomena, e.g., for heat transfer define ‘Heat Flux’ or for a stress field define ‘Force’. This abstraction enforces the localised context for each component. For a component producing a heat field, it needs only output a ‘Heat Flux’ parameter value to its boundaries. It is up to each boundary to present this effect to the context of the neighbouring component in a suitable form. Thus the context of a neighbour will incorporate ‘Heat Flux’ which will be presented as
`Heat Influx` parameter with a set value. From the receiving components perspective, the origin of the heat is immaterial as are the destinations for subsequent side-effects. A boundary link will be represented as an object with initial attributes of shape and participating components.

3. Related Work

While there has been considerable investigation of different types of abstract models for physical systems, the problem of modelling physical phenomena in such systems has received little attention. However, work has been done which is related to parts of the proposed modelling process.

Several artificial intelligence researchers have investigated the use of qualitative reasoning to predict the behaviour of physical systems. Section 4 explores the contribution of this work in greater detail. The proposed system is a modelling process directed to modelling physical phenomena whereas the applications of existing qualitative reasoning systems have focussed on predicting device behaviour, e.g., [Forbus et al `87].

Geometric idealisation plays an important role in engineering modelling. [Baehman et al `88], [Baehman `88] and [Collar `90] have done considerable work in the field of geometric modelling but the starting point for their idealization process is a lower level, fully specified physical model. In contrast, the proposed system begins with a high-level, incomplete, abstract specification (i.e., the complex model) where the behaviour of the system is not yet determined. Nonetheless, the geometric idealization techniques proposed in their research is valid in the lower levels of the proposed architecture specifically at phenomena and analysis volume reduction stages.

Context dependent behaviours [Nayak et al `90] represent a conceptually appealing method for organising multiple models of primitive components. Unfortunately, like component based and process based qualitative reasoning systems, this ontology is geared towards prediction of device behaviour and has limited application in modelling physical phenomena.

[Geese `90] describes a program for automated physical modelling. He proposes a quantitative modelling approach whereby the behaviour of the system (a mechanical device) is derived using numerical simulation. Again, the application for this work is in the area of kinematic analysis rather than physical phenomena. Furthermore, the use of quantitative techniques requires a more complete model specification than is provided for the user interface of the proposed system.

4. Qualitative Reasoning

Most of today's expert systems have invariably modelled their domain with a "black box" approach, i.e., compiling rules from observable inputs and outputs, with no consideration for underlying physical mechanisms. A commonly recognised failing of such "shallow models" is that they are highly domain specific [Kuipers `86]. The black box approach to knowledge representation is a shallow way of contriving summary rules to suit the needs of the application which must use them.

Qualitative reasoning provides for "deeper" knowledge which models the underlying systems of the black box and allows the derivation of the high level rules of input-output by applying an input to the model and allowing simulation to predict the output. In contrast to shallower conventional heuristic models, a qualitative reasoning system will embed its knowledge in real physical mechanisms rather than in rules of the "mb based on intuition and experience.

In this section three approaches to qualitative reasoning are discussed. In particular abstract qualitative reasoning systems are defined and how principal components are instantiated within component based, process based and constraint based systems respectively. Furthermore, the relevance of these methodologies in the context of the proposed architecture is discussed (Figure 1 Transition B).

4.1 Components of a qualitative reasoning system.

A qualitative reasoning method begins with a model of the domain, called a structural description, in which precise numerical values and precise functional relationships are absent. All parameters can take on one of a finite set of non-numeric values which generally define discrete regions of qualitatively distinct behaviour (e.g., fluid flow rate could be expressed as being laminar, between laminar and turbulent, turbulent). An ordered set of such qualitative values is called a quantity space. The minimal quantity space has the values \(\{+, 0, -\}\) which only supports a system to reason about signs.

The influence of one parameter on another is expressed by qualitative constraint equations. The qualitative state of the system is an assignment of a qualitative value and incremental Qualitative value or IQ value (i.e., direction of change which may be specified as decreasing, steady, or increasing) to each parameter together with the time point or interval at, or over, which that value applies and the direction in which the parameter value is changing (if at all). Time is represented as an ordered sequence of time-points generated dynamically whenever something interesting happens to a parameter. Transition rules govern any changes in parameter value.

Some systems based on qualitative reasoning use simulation as their inference procedure, while others use envisionment. Simulation begins with an initial state and propagates values forward, exploring all legal possibilities, until no more transition rules apply. Envisionment generates all legal qualitative states and then all possible transitions between these states. Both approaches result in a graph of possible states with arcs signifying permissible transitions between states. A qualitative behavioural description is then any path through such a graph.

So any qualitative reasoning system can be characterised by its structural description, quantity space, notion of qualitative state representation of time, transition rules, behavioural description and inference procedure. The major difference between the various approaches to qualitative reasoning concerns the precise statement of the constraint laws and how they are derived from the physical structure [De Kleer and Brown `84]. A central organising principle for de Kleer and Brown [De Kleer and Brown `84] is the notion of component, for Forbus the notion of process [Forbus `84] and for Kuipers the notion of constraint [Kuipers `86]. The preceding views of parameters, constraints, qualitative state and representation of time are essentially shared among the qualitative physics of component based, process based and constraint based systems.

4.2 Envisioning

De Kleer and Brown take the view that a device consists of physically distinct parts connected together. The goal is to draw inferences about the behaviour of the composite device solely from laws governing the behaviour of parts. Their central modelling principle is that the qualitative differential equation, called a confluence, which acts as a constraint on the variables and derivatives associated with components. For example, consider the qualitative behaviour of a pressure regulator (Figure 3) expressed by \(\frac{dP}{dt} + A \frac{dQ}{dt} = 0\), where \(P\) is the pressure across the valve, \(Q\) is the flow through the valve and \(A\) is the area available for flow. \(dP\), \(dQ\) and \(A\) represent the changes in \(P\), \(Q\) and \(A\) respectively. The confluence represents multiple competing influences, the change in area positively influences flow rate and negatively influences pressure. The change in pressure positively influences flow rate, etc. A confluence is generally valid for a certain operating range of some component. In this example very different behaviours occur when the valve is fully open or fully closed.

4.3 Qualitative Process Theory

Forbus describes a physical situation in terms of the interaction of competing processes [Forbus `84]. The central idea is that all changes in physical systems are caused directly or indirectly by processes. Processes can become active or inactive as parameter
The physical situation presented in Figure 4 can be described by the Kuipers' constraint based approach and conforms most readily to the process of predicting a composite device behavior. This work is more concerned with the behavior of the system as a whole, rather than each component having a behavior itself. Instead of predicting a composite device behavior, this work is more concerned with the behavior within a component, i.e., physical phenomena. Rather than a component having a behavior itself, in the approach taken here, it serves more as a site for behaviors to occur. The composite behavior in the modeling approach of the proposed system looks at how physical phenomena, local to one site (component) initially, might propagate to other sites (components) in the system.

4.4 Qualitative Simulation

In Kuiper's constraint-based approach, the constraints on how parameters are related to each other are two- or three-place relations on physical parameters. Some specify familiar mathematical relationships: DERIV (velocity, acceleration), MULT (mass, acceleration, force), MINUS (forward, reverse). Others assert qualitatively that there is a functional relationship between two physical parameters, but only specify that the relationship is monotonically increasing or decreasing. M+(age, experience), M- (mpg, mph). Inequality and conditional constraints specify conditions under which some constraint holds.

The "causal structure description" indicates each of the constraints and parameters of the model. Consider the simple physical system in Figure 4 consisting of a closed container of gas (at temperature Ta) that receives heat from a source (Ts) and radiates heat to the air. The rate of flow of heat into the gas is a strictly increasing function of the temperature difference between the gas and the source. dT = 0 corresponds to no heat flow into the gas. To solve this model, a qualitative simulation is carried out by propagating +0,- values (using Figure 5) and inequalities (using constraint propagation) in order to obtain values for all the variables.

4.5 Qualitative Reasoning for Modelling Physical Phenomena

De Kleer and Brown's component-based reasoning system determines a composite device behavior from its component structure. Domains consist of specific components which have their structural description mapped out explicitly in the form of confluenes (qualitative differential equations). The focus for behavior prediction rests on the behavior of the device itself rather than on underlying internal effects.

A similar observation applies to Forbus' qualitative process theory. The central idea of a process as the essential agent for change in a physical system fits well with the notion of physical phenomena introducing side-effects in other components. However, this approach is also applied to predicting device behavior [Forbus et al '87].

4.6 Proposed Modelling Approach

The modelling approach proposed in this work deals with behavior at a lower level than component and process based systems. Instead of predicting a composite device behavior, this work is more concerned with the behavior within a component, i.e., physical phenomena. Rather than a component having a behavior itself, in the approach taken here, it serves more as a site for behaviors to occur. The composite behavior in the modelling approach of the proposed system looks at how physical phenomena, local to one site (component) initially, might propagate to other sites (components) in the system.

Kuipers' constraint-based approach conforms most readily to the requirements of this work. Constraint equations can be adapted to qualitatively reflect fundamental laws of physics, e.g., power dissipation = M+(voltage). The calculus of qualitative simulation will allow side-effects of physical phenomena to be derived. Parameters provide an ideal method for propagating effects by parameter sharing between neighboring components.

In contrast to all three qualitative reasoning systems outlined in this section, it's not of interest to predict system behavior over time. Instead, the proposed system incorporates the qualitative calculus and constraint conventions of Kuipers' simulation-system to perform dynamic simulation but present the resulting behavioral description as a static view of the final time-point parameter values. The graph of transitions may be used as a historical account of the simulation to justify results to the user.

5. Qualitative Causal Networks

Reason locally and propagate globally: this is the basic principle from which the qualitative causal network structure is derived as a tool for modelling physical systems. In this section the use of qualitative simulation is explained to determine local behavior at each component (site) in the network and how a causal network, implemented through boundary links between neighboring components, is used to propagate side-effects to 'connected' components.

The simulation process described in Section 5.1 corresponds to A and B of Figure 6 which outlines the algorithm used in the proposed system to generate a behavioral description of the physical system (Figure 1, Stage 3). Section 5.2 examines how causal networks are used to propagate side-effect phenomena to the contexts of neighboring components (Figure 6, Part C).
The basis of the network structure is the boundary links between neighbouring physically connected components. The boundary is characterised by its participant components, its geometry (which may determine the type of distribution for a phenomenon), and two lists of transfer parameters, (i.e., source A <-> destination B and source B <-> destination A).

After a local simulation has been completed, the boundary link’s outgoing transfer parameters are updated either by appending a new parameter or by changing a current transfer parameter’s value and/or IQ value. The next component incorporates all incoming transfer parameter lists into its context. Simulation then proceeds with the context as an initial state.

5.3 Stopping Criteria

From Figure 6 it is clear that the simulation process is an iterative one. One complete iteration corresponds to a local simulation/global propagation cycle. The behavioural description is considered to be complete when the system reaches an equilibrium. In practice, a physical equilibrium will be represented by a complete iteration which produces no new states of parameters. Each component’s active parameter list should be empty at the start of such an iteration.

The final behavioural description (Figure 1: Stage 3) will be determined by the values of key parameters in each component and in each boundary’s transfer lists.

6. Conclusions

In this paper a conceptual architecture for modelling abstract-physical systems has been outlined. In particular qualitative causal networks have been introduced as a tool for modelling physical phenomena. It has been demonstrated how this network structure facilitates the presentation of a localised perspective of the whole system to each component. Boundary links are used to present summary effects to a component allowing qualitative simulation to be done without reference to the source or destination of side-effects. This is an important division of work from the design point of view. Causal network management and qualitative simulation can be developed as self-contained, independent modules.

A role for qualitative simulation in modelling physical phenomena has been established. The importance of deep knowledge for intelligent reasoning systems has been emphasised in this architecture. The need for shallow rule-based inference systems is also recognised and incorporated into the proposed system to initialise the domains for qualitative simulation and finally in model optimisation.

Having outlined the overall modelling process, the prototype implementation will begin with the primary simplification and concentrate on the derivation of a behavioural description in the manner outlined in Section 5.

Acknowledgements

This work is sponsored by the Hitachi Dublin Laboratory. We would like to thank Nobutoshi Sagawa and Neil Hurley for various discussions on the subject and comments on earlier drafts of this paper.

References


SOLIDIFICATION OF VARIABLE PROPERTY MELTS IN CLOSED CONTAINERS: MAGNETIC FIELD EFFECTS

George S. Dulikravich  
Associate Professor  
Aerospace Eng. Dep.  
Penn State University  
University Park, PA

Branko Kosovic  
Graduate Student  
Aerospace Eng. Dep.  
Penn State University  
University Park, PA

Sanseung Lee  
Research Scientist  
Research Scientist  
Aerospace Engineering Department  
Agency for Defense Development  
Daejon, South Korea

Abstract: A computer code has been developed for the numerical prediction of steady, laminar, incompressible flows with strong heat conduction, magnetic field effects (Lorentz forces and Joule heating), latent heat of phase change, and thermal buoyancy using extended Boussinesq approximation. The same code predicts the fluid flow field and the solid layer resulting from strong wall cooling. Numerical results for solidification inside a closed container demonstrate the influence of strong magnetic fields on the melt flow field and the solidification interface geometry.

I. INTRODUCTION

Based on our earlier works [1-3] in computational magnetohydrodynamics (MHD) for steady, laminar, incompressible flows in two and three dimensions, we have recently developed a computer code that is capable of simultaneously predicting details of the melt flow field and the formation of the solidified region [4].

Boussinesq approximation was used to account for the thermal buoyancy force, while allowing the coefficients of viscosity, heat conduction, and specific heat to depend on temperature arbitrarily [5]. Nevertheless, in the present work, values of these coefficients were kept constant within the liquid, allowed to vary linearly between liquids and solidus temperatures, and then again kept constant within the solid. A special test run (b) was performed where the coefficient of viscosity was varied according to the arctangent law over the entire range of temperatures. This computing logic enables us to use a single flow field analysis code in order to simultaneously predict both the fluid flow field and the temperature field inside the accreting solid, thus capturing the solid/liquid interface shape without any special front tracking algorithm.

II. ANALYTICAL MODEL

Navier-Stokes equations for incompressible electrically conducting homopropotional fluid flow are given by

\[ \nabla \cdot \mathbf{u} = 0 \]  
\[ \nabla \cdot \left( \mathbf{u} \cdot \mathbf{B} \right) = 0 \]  
\[ \rho \nabla \cdot \mathbf{u} = \nabla \cdot \left( \mu \nabla \mathbf{u} + \eta \mathbf{u} \right) \]  
\[ \rho c_p \nabla T = k \nabla^2 T - \mathbf{u} \cdot \mathbf{J} \]  
\[ \mathbf{J} = \sigma \mathbf{E} \]  
\[ \mathbf{E} = \nabla \mathbf{B} \times \mathbf{B} \]

where \( \rho \) is the density, \( \mathbf{u} \) is the velocity, \( \mathbf{B} \) is the magnetic field, \( \mu \) and \( \eta \) are the dynamic and kinematic viscosity coefficients, \( k \) is the thermal conductivity, \( c_p \) is the specific heat, \( T \) is the temperature, \( \sigma \) is the electric conductivity, and \( \mathbf{J} \) and \( \mathbf{E} \) are the electric current density and electric field, respectively.

The system of governing partial differential equations (1-4) was iteratively integrated using explicit Runge-Kutta four-stage time stepping, and an artificial compressibility formulation [1], except that the velocity components and their derivatives were explicitly set to zero at every point where the instantaneous temperature is lower than the solidus temperature.

III. RESULTS

In a conventional case of a thermal buoyancy induced flow inside a closed rectangular container, the bottom wall is uniformly hot and the top wall is uniformly cold, while the vertical walls were thermally insulated. The computed velocity vector field [2] for such a test case is depicted in Figure 1a. If a uniform vertical downward-pointing magnetic field is added (but without allowing for solidification), the number of recirculating flow regions will change (Fig. 1b), indicating strong influence of the magnetic field on the flow pattern, that is, the reduction of vorticity.

To demonstrate the capability of the code to predict the formation of the solid region, the top of the container was uniformly undercooled. The solid phase was predicted to grow from the top wall (Fig. 2). Figure 2 shows the velocity vector fields for the cases with: \( \mathbf{H} = 0 \) (Fig. 2a), \( \mathbf{H} = 0 \) with variable viscosity (Fig. 2b), \( \mathbf{H} = 5 \) (Fig. 2c), and \( \mathbf{H} = 10 \) (Fig. 2d). It can be seen that with the increase of the strength of the magnetic field, the vorticity diminishes and the recirculating cell patterns change. Isotherms for the same sequence of test runs are shown in Figure 4. The convergence histories for this test case are represented in terms of an instantaneous count of the solidified cells (Fig. 3). The convergence is oscillatory with a clear indication that the change in the character of the temperature field is smoothly influenced by the magnetic field.

In summary, the code has been tested for various test cases with different magnetic field strengths. The code is capable of predicting the formation of the solid region and the effect of magnetic fields on the flow field. The results indicate that the magnetic field can significantly influence the flow field and the solidification process.

IV. REFERENCES


**Figure 1.** Recirculation streamlines inside the closed container without solidification and with $Pr = 7.9$, $Gr = 3000$, $Re = \sqrt{Gr}$, $Ec = 1$ for: a) $Ht = 0$; b) $Ht = 5$.

**Figure 2.** Velocity vector field inside the melt with $Pm = 1$, $Pr = 7.9$, $Gr = 3000$, $Ec = 1$, $Re = \sqrt{Gr}$ and: a) $Ht = 0$; b) $Ht = 0$ and variable viscosity; c) $Ht = 5$; d) $Ht = 10$.

**Figure 3.** Convergence histories: number of solidified cells versus number of iterations.

**Figure 4.** Isotherms inside the solid and the melt with $Pm = 1$, $Pr = 7.9$, $Gr = 3000$, $Ec = 1$, $Re = \sqrt{Gr}$ and: a) $Ht = 0$; b) $Ht = 0$ and variable viscosity; c) $Ht = 5$; d) $Ht = 10$. 
AN ASYMPTOTIC MODEL FOR DETONATION IN DUCT

S. GERBI
ENS Lyon, 46 allée d'Italie, 69364 LYON CEDEX 07 France

G. CLAUS
ELF-CRES, Chemin du Canal, BP 22, F 69360 ST SYMPHORIEN DOZON, FRANCE

Abstract: An asymptotic model for a detonation in duct is derived through activation energy asymptotics. Using linearized stability analysis, we study the stability of the constant equilibrium state; next we investigate the quenching phenomenon. Finally we study a bifurcation phenomenon related to the length of the duct and propose a simplified model.

1. Introduction

The steady detonation structure consists of a shock wave, followed by an induction zone in which reaction is weak; followed by a zone in which vigorous reaction and heat release occurs. Behind the reaction zone is uniform burnt gas. In the limit of large activation energy (\(g \to \infty\)) this structure reduces to the well known square wave in which conditions are uniform in the induction zone (of length \(L\)) and the reaction zone is a discontinuity.

When the stability of such detonation wave is examined, for a certain class of disturbances based upon \(L\) and \(g\), [1], [2] perturbations to the shock displacement are governed by the fully nonlinear parabolic equation

\[
\begin{align*}
g_t + \frac{1}{2} g^2 g_x = & \ln \left( \frac{e^{2g_{xx}} - 1}{c e^{g_{xx}}} \right) \\
g_x(0,t) = g_x(L,t) = & 0
\end{align*}
\]

associated with the natural Neumann boundary conditions at the walls:

\[
(2)\quad g_x(0,t) = g_x(L,t) = 0
\]

in which \(c\) is a positive given constant and \(L\) is the width of the duct. In this presentation, we are interested in the stability of steady-state solutions of problem (1)-(2). Of course \(g \equiv 1\) is the expected unperturbed solution but numerical investigation has shown the existence of periodic non-constant solutions for a certain range of parameter \(L\). Two kinds of results have been obtained for the evolution of a perturbation of the constant solution \(g \equiv 1\).

2. Linearized stability analysis

Linearizing problem (1)-(2) around \(g = 1\), we get the dynamical system (3)-(4)

\[
\begin{align*}
u_t + u u_x = & \frac{c}{2} u_{xx} + u \\
v_x(0,t) = v_x(L,t) = & 0
\end{align*}
\]

The spectrum of \(\mathcal{L}\) consists of the eigenvalues \(\lambda_j = 1 - \frac{j^2 \pi^2}{2L^2}\) with corresponding eigenvectors \(\omega_j = \cos(j \pi x), j = 0, 1,\ldots\).

There is at least one unstable mode (the planar mode) corresponding to \(\lambda_0 = 1\). We define the critical length \(L_c = \pi \sqrt{\frac{c}{2}}\) for which \(\lambda_1 = 0\), leading to a center manifold. For \(L < L_c\), the stable manifold is codimension 1. Numerically, we validate this behaviour on the nonlinear problem (1)-(2) by taking a small initial condition with null mean-value: the evolution of such a perturbation vanishes [4].

3. Quenching phenomenon

This result is also related to the previous stability analysis: taking a non periodic initial condition leads to an unstable behavior. Nevertheless, the solution remains bounded whereas the time derivative \(g_t\) blows-up in finite time \(t_c\): this is the so-called quenching phenomenon.

More precisely, an analytical argument [4] shows that:

\[
g_t - (t - t_c) \log(t - t_c)
\]

4. Critical length \(L_c\) and bifurcation phenomenon

As previously mentioned, there exists a critical length \(L_c\) for which \(\lambda_1\) crosses 0. Therefore as \(L\) increases, crossing \(L_c\), the dimension of the stable manifold decreases. On the other hand, we exhibited nontrivial solutions for \(L > L_c\). Figure 1 gives a description of the stability-state periodic solutions in the plane \((\sqrt{\frac{2}{c}}L, g(0))\).
This is the typical exchange of stability occurring when there exists a branch of bifurcation. In order to give a mathematical description of this bifurcation phenomenon, we derive a simpler model whose behavior can be compared to the physical one. Assuming \( |g| \) and \( |g_{xx}| \) to be small and taking a first order approximation of the logarithmic term we get:

\[
(5) \quad u_t - u_{xx} = \frac{2f^2}{c} (1 - u) \log (1 - u)
\]

\[
(6) \quad u_x (0,0) = u_x (0,1) = 0
\]

where \( u \) is defined by:

\[
(7) \quad g = 1 - c \log (1 - u)
\]

On this model, we study the existence of non-trivial stationary solutions and their stability as well as the quenching phenomenon.

References:


Homogenization for nonlinear adsorption-diffusion processes in porous beds

Éric Canon
Université de Saint-Étienne
23 rue du docteur Paul Michelon
F-42023 Saint-Étienne Cédex 2.

Abstract: A homogenized model for nonlinear adsorption-diffusion processes in porous beds, modelling chromatographic columns, is proposed. We consider an inhomogeneous periodic medium with two levels of structure having strong different scales. This suggests us to homogenize it. The first structure is a convection area. The second structure is an adsorption area, made of small porous crystals. Our equations are convection-diffusion equations coupled with Stokes equations for the fluid velocity. Along the crystal boundary, we have three kinds of discontinuity: jumps of the fluid velocity and of the diffusion coefficient, and a nonlinear jump for the concentration. This last heterogeneity is the most important in view of homogenization. We show a maximum principle ans some energy estimates for our nonlinear problem. From them, we prove, as main result, the convergence of our microscopic model to a homogeneous nonlinear macroscopic model.

I. INTRODUCTION

This paper presents a general homogenized macroscopic model for convection-diffusion equations into a chromatographic column. This model takes into account the complete internal heterogeneous structure of the column, and nonlinear relationships between moving and adsorbed phases. The main point of our problem is this nonlinearity, this discontinuity is an isothermal relationship derived from thermodynamical considerations. A chromatographic column is constituted, at a microscopic scale, of two structures having strong different scales: a convection area and an adsorption area made of small porous crystals. These structures are assumed to be periodic. Our equations are convection-diffusion equations coupled with Stokes equations. The difference of scales between the two levels suggests us to use the homogenization techniques to replace the heterogeneous medium by an equivalent homogeneous medium. The structures are assumed to depend on some small parameter $\varepsilon$. This parameter will tend to zero in the homogenization process. We shall show that under some reasonable assumptions on the isothermal jump condition, the model converges to some nonlinear integrodifferential equation. The $\varepsilon$-depending model will be called the microscopic model. The homogeneous limit model will be the macroscopic model.

II. THE MICROSCOPIC MODEL

Our equations are convection-diffusion equations coupled with Stokes equations. The heterogeneities are: jumps of discontinuity for the velocity and for the diffusion coefficient, and a nonlinear jump for the concentration. The last one is the most important. Our domain $\Omega$ is divided into two parts $\Omega_1^e$ and $\Omega_2^e$, corresponding to the inhomogeneous parts of the column. $\Gamma^e$ denotes the boundary between $\Omega_1^e$, $\Omega_2^e$. The unknown function is the corresponding concentration denoted by $w_i^e$, $i=1, 2$. The fluid velocity is denoted by $u^e$, and the coefficient of diffusion by $D_i$, $i=1, 2$. The geometry of $\Omega_1^e$, $\Omega_2^e$, $\Gamma^e$, and consequently the solutions $w_i^e$, and $u^e$ depend on $\varepsilon$. We have to homogenize the following microscopic model:

$\nabla \cdot w_i^e = D_i \Delta w_i^e - u^e \nabla w_i^e$ (1)

$\nabla \cdot w_2^e = \varepsilon^2 D_2 \Delta w_2^e$ (2)

$w_2^e = h(w_1^e)$ on $\Gamma^e$ (3)

$\nabla u^e \cdot n = 0$ on $\Gamma_{in}$ (4)

$\nabla u^e \cdot n = 0$ on $\Gamma_{out}$ (5)

$\int_{\Omega_i^e} w_i^e \cdot n = 0$ on $\Gamma_{in}$ (6)

$\int_{\Omega_i^e} u_i^e \cdot n = - \int_{\Omega_i^e} u_i^e \cdot n$ (7)

$\Delta u^e = \nabla \cdot u^e$ (8)

$\div u^e = 0$ (9)

with the following boundary conditions:

$u^e = 0$ on $\Gamma^e \cup \Gamma^0$ (and in $\Omega_2^e$) (10)

$u^e = 0$ on $\Gamma_{in} \cup \Gamma_{out}$ (11)

$\int_{\Omega_i^e} u_i^e \cdot n \leq 0$ on $\Gamma_{in}$ (12)

$\int_{\Omega_i^e} u_i^e \cdot n = - \int_{\Omega_i^e} u_i^e \cdot n$ (13)

III. THE MACROSCOPIC MODEL

In this section we give the macroscopic model which is defined to be the limit model of the microscopic model stated in the previous section when, the parameter $\varepsilon$ goes to zero, the convergence will be examined in the next section. Notice that this model is uniform for the whole domain $\Omega$. The macroscopic model is defined as follow (where $\bar{W}$ denote the concentration in $\Omega$ and $\bar{u}$ the fluid velocity):

$\int_{\Omega} \partial_t \bar{w} + f(t) * \partial_j h(\bar{w}) - \sum_{i,j=1}^3 a_{ij} \partial_i \partial_j \bar{w} + \bar{u} \cdot \nabla \bar{w} = 0$ (14)

The deformation of the Laplace operator due to the geometry of the heterogeneity is given by:

$a_{ij} = D_1 \int_{\Omega_1} \partial_j h(\bar{w}) + \frac{1}{\varepsilon} \int_{\Omega_1} \nabla \sigma_k(y) dy$ (15)

where $\sigma_k$ is a $Y$-periodic function in $H^1(Y_1) (Y$ is a basic cell, $Y_1$ is the convection part of the cell $Y$) defined as:

$\Delta \sigma_k = 0$ in $Y_1$ (16)
The convolution term \( f \) expressing the adsorbed phase is given by:
\[
f(t) = \frac{1}{\mathcal{Y}} \int_{\mathcal{Y}_2} \rho_0(t,y) \, dy
\]
where \( \rho_0 \) is a \( \mathcal{Y} \)-periodic function in \( H^1(\mathcal{Y}_2) \) and \( \mathcal{Y}_2 \) is the adsorption part of the cell \( \mathcal{Y} \) defined as:
\[
\begin{align*}
\partial_t \rho_0(y,t) - D_2 \Delta \rho_0(y,t) &= 0 \text{ in } \mathcal{Y}_2 \\
\rho_0(y,t) &= 1 \text{ on } \mathcal{Y}_1 \cup \partial \mathcal{Y}_2 \\
\rho_0(y,0) &= 0
\end{align*}
\]

The fact that the adsorbed phase appears in (14) as a memory term is strongly related to the fact that equation (2) in the microscopic model is linear. For a more general model, we get a coupled system in place of equation (14): a convective-diffusive equation for the whole domain \( \Omega \) with a source term for the adsorption process, coupled with a diffusive cell equation in \( \mathcal{Y}_2 \) involving \( h \) on the boundary \( \partial \mathcal{Y}_2 \).

**IV. CONVERGENCE**

It is well-known (Tartar) that the homogenization of the Stokes' equations in porous media leads to the Darcy's law. We do not rewrite it here.

We assume the following properties for some extension of the boundary condition \( \nu_0 \) in (5):
\[
\begin{align*}
\nu_0 \in L^2(0,T;H^1(\Omega)) \cap H^1(0,T;L^2(\Omega)) \cap L^\infty([0,T] \times \Omega) \quad (22) \\
n \nu_0 \in L^2(0,T;H^1(\Omega)) \cap H^1(0,T;L^2(\Omega)) \cap L^\infty([0,T] \times \Omega) \quad (23)
\end{align*}
\]
The convergence of the microscopic model to the macroscopic model is obtained for the weak formulation of the models given in the two previous sections. It follows of the a priori estimates stated in the two following lemmas:

**Lemma 1 (maximum principle):**

Under hypothesis (22) and (23) the following estimates hold for the solution \( w^\varepsilon \) of the microscopic problem:
\[
0 \leq w^\varepsilon(x,t) \leq \sup (\nu_0(x,t) ; (x,t) \text{ in } \Gamma_{IR} [0,T]) \text{ for almost every } (x,t) \text{ in } \Omega \times [0,T].
\]
\[
0 \leq \partial_t w^\varepsilon(x,t) \leq \sup (\partial_t \nu_0(x,t) ; (x,t) \text{ in } \Gamma_{IR} [0,T]) \text{ for almost every } (x,t) \text{ in } \Omega \times [0,T].
\]
The proof is based on Stampacchia's method.

**Lemma 2 (energy estimates):**

For the solution \( w^\varepsilon \) of the microscopic problem the following norms are bounded independently of \( \varepsilon \):
\[
\begin{align*}
\|w^\varepsilon\|_{L^2(0,T;L^2(\Omega_x))}, \|\nabla w^\varepsilon\|_{L^2(0,T;L^2(\Omega_x))}, & \|\nabla^2 w^\varepsilon\|_{L^2(0,T;L^2(\Omega_x))}, \\
\|\partial_t w^\varepsilon\|_{L^2(0,T;L^2(\Omega_x))}, \|\partial_t \nabla w^\varepsilon\|_{L^2(0,T;L^2(\Omega_x))}, & \|\partial_t \nabla^2 w^\varepsilon\|_{L^2(0,T;L^2(\Omega_x))}.
\end{align*}
\]

Let us define \( \overline{w}^\varepsilon \) the h-harmonic extension of \( w^\varepsilon \):
\[
\Delta h (\overline{w}^\varepsilon) = 0 \text{ in } \Omega_x^\varepsilon.
\]

We have the following convergence result:
A PHASE FIELD MODEL FOR ISOTHERMAL SOLUTION GROWTH

A. A. WHEELER
National Institute of Standards and Technology
Gaithersburg, MD 20899 U.S.A.

and

W. J. BOETTINGER
National Institute of Standards and Technology
Gaithersburg, MD 20899 U.S.A.

ABSTRACT — We describe a new phase field model for the phase transition of an isothermal binary alloy. This is the first time, to the authors knowledge, that a phase field model has been proposed for phase transition in a impure material. This represents a significant step in the derivation of a phase field model for the solidification of a nonisothermal binary alloy.

I. INTRODUCTION

Classical macroscopic models of phase transitions model the interface between regions of different phase as a surface, and hence assume it has zero thickness. The governing equations for thermodynamic variables, such as temperature and solute, are formulated in each phase independently, based upon conservation principles and quantitatively verified phenomenological laws. The boundary conditions at the interface are chosen to describe the processes, such as liberation of latent heat and segregation that occur at the interface on a microscopic scale. This approach gives rise to the formulation of a free boundary problem which provides a difficult mathematical setting and only the simplest models of phase change have been rigorously mathematically analyzed. The advantage of these models is that it is clear from the outset what physical mechanisms are incorporated into them, and comparison with careful controlled experiments is possible.

An alternative technique for investigating transport processes in systems involving a phase transition, involves the construction of a Landau-Ginzberg free energy functional. This approach has its roots in statistical physics, Landau and Khalatnikov [1]. Further, a phase field, which is a function, \( \phi(x,t) \) is postulated which describes the phase of the system at any point in time and space. It is assumed that the Helmholtz free energy \( G(\phi, t) \) is a functional of the phase field, as well as any other thermodynamic variables, (such as temperature which are denoted here by ellipsis) in the following way:

\[
G(\phi, \ldots) = \int_{\Omega} \left( \frac{1}{2} \kappa (\nabla \phi)^2 + g(\phi, \ldots) \right) \, d\Omega, \tag{1}
\]

where \( \Omega \) is the region occupied by the system, and \( g(\phi, \ldots) \) is the free energy density. Its dependence on \( \phi \) usually has a "double well" form. The phase field is then assumed to evolve as:

\[
\dot{\phi} = L \left( \frac{\delta G}{\delta \phi} \right), \tag{2}
\]

where \( L \) is some partial differential operator. This equation is then supplemented by partial differential equations for the other thermodynamic variables. Cahn, [2], has successfully used this approach to model spinodal decomposition of a binary alloy, although how the concentration naturally plays the role of the phase field. Various models that employ this idea are reviewed by Halperin, Hohenberg and Ma, [4], particularly in regard to the study of critical phenomena. The Model C given by these authors has been adapted by Langer, [5], Fix, [6] and most prolifically by Caginalp, [7] to derive the so-called "phase-field model" of solidification which models the phase change of a pure material. Caginalp has extensively studied this, and variations of this model, [8] [9]. It has emerged from study of this model that qualitatively it exhibits features common to solidification of a pure material. Numerical calculations based on this model, by Smith, [10], and a similiar model, by Kubyshev [11], show breakdown of a planar and circular interfaces by cellular structures, as well as the formation of dendrite like structures, liquid trapping and coarsening behaviour.

Caginalp, [8], has shown in various distinguished limits, in which \( \epsilon \to 0 \), that various forms of the classical Stefan problem may be recovered, in which the interface is taken to be "sharp" i.e. modeled by a surface. In this limit there are thin layers within \( \Omega \) of thickness \( \mathcal{O}(\epsilon) \) in which the phase field rapidly changes. These are interpreted as representing interfaces, which are necessarily diffuse. From this analysis it transpires that in some limits, the interfacial dynamics involve curvature effects corresponding to the Gibbs-Thompson interfacial surface energy as well as kinetic effects. Further, it is also possible to recover the classical Hele-Shaw problem in other limits. It is clear that this approach can embody a considerable variety of realistic physical effects in a coherent way.

However, this superabundance of physical phenomena also provides a difficulty when applying the model to a definite physical situation. This is because it is not clear how to choose the values of parameters in the phase field model so that it models the solidification of a pure material with given materials and growth parameters, (or equivalently the Stefan number and capillary number).

Another difficulty with this particular model, as pointed out by Penrose and Fife [12], is that it is thermodynamically inconsistent. This is because the free energy functional is only employed in the formulation of the kinetic equation for the phase field. The concern here is that the solution of the above governing equations does not correspond to the free energy decreasing monotonically with time, as required by the Second Law of Thermodynamics. An alternative approach suggested by these authors is to construct an entropy functional, \( S \), of the system and require it to evolve as a gradient flow of the form:

\[
\dot{u} \propto \text{grad}_u S(u), \tag{3}
\]

where \( \text{grad}_u \) is a suitable constrained gradient, and \( u \) represents the thermodynamic variables. This formulation necessarily ensures that the total entropy of the system increases with time.

The appeal of phase field models in describing phase transitions is twofold;

- It provides a simple, elegant description, that appears to embody a rich variety of realistic physical phenomena.
- From a computational point of view it is relatively simple to compute solutions. This is because it is not necessary to distinguish between the different phases. Computations on the classical sharp interface formulation require that the free boundary is tracked numerically and that the region occupied by each phase is therefore determined and dealt with individually. This results in very difficult and unidy numerical algorithms.

In this paper we derive a new phase field model for phase transitions of an isothermal binary solution. To our knowledge, there are to date no phase field models that deal with impure materials. The model presented here is a first step to developing a phase field model for the solidification of an alloy.

II. PHASE FIELD MODEL

We consider an isothermal solution of two different species \( A \) and \( B \) in which are present two phases, solid and liquid, contained in a fixed region \( \Omega \) with boundary \( \partial \Omega \). We denote the concentration of \( B \) by \( \epsilon A \), and we introduce a phase field \( \phi(x,t) \) which represents the phase in time and space in \( \Omega \). For definiteness we describe the
solid liquid interface by φ(x,t) = ½ and denote solid regions where φ(x,t) > ½ and liquid regions where φ(x,t) < ½.

A recent phase field mo- del due to Kobayashi [11] models the phase transition of a pure material by employing the following gradient weighted free energy functional:

\[ F_\varepsilon(\phi, T) = \int_0^2 \left| \nabla \phi \right|^2 + f(\phi, T) \, d\Omega, \]

where ε is a constant, T(x,t) is the temperature and the free energy density f(\phi, T) is represented by:

\[ f(\phi, T) = f^I + \varepsilon \int T \left[ -1 \beta(T) - \frac{1}{\varepsilon} - \beta(T) \right] \, dp, \]

where β(T) is a monotonic increasing function of T, such that β(T_m) = 0, where T_m is the freezing temperature of the material and |β(T)| < ½. The free energy density f(\phi, T) is a double minimum potential. The restriction |β(T)| < ½ ensures that it has local minima at φ = 0 and φ = 1, and local maxima at φ = ½ + β(T). Because of the two minima the system may exist stably in a state which is all-liquid (φ(x,t) = 0) or all-solid (φ(x,t) = 1). There is an energy penalty for a change of phase within the region Ω, which corresponds to φ varying between zero and unity. This is because such a variation increases the total energy of the system, due to an increased energy density associated with the double well nature of the energy density, and also due to the contribution to the total energy due to the gradient energy, which no longer zero.

If \(-\frac{1}{\varepsilon} < \beta < 0\), then the global minima of the energy density is at \(\phi = 1\) and so the all-solid state is the one with the lowest energy and is the preferred state. However, if 0 < β < \(-\frac{1}{\varepsilon}\) then the situation is reversed and the liquid is the preferred state. We see that at temperatures below the melting point the solid phase has the minimum energy and is preferred, whereas for temperatures above the melting temperature, the all-liquid phase is preferred.

We now employ this form for the free energy density to develop the appropriate free energy density for an isothermal ideal solution. We assume that the temperature, T, which is given, is such that if the solution consisted only of species A (c = 0) the all-solid phase would be the preferred state i.e. T < T_m, where T_m is the melting temperature of pure A. Further, we also assume that the temperature T is sufficiently large that if the system consisted of species B alone (c = 1) the all-liquid phase would be the preferred state i.e. T > T_m, where T_m is the melting temperature of pure B. We also assume that the molar Gibbs free energy densities of each species A and B alone are of the form given by Kobayashi, and are denoted by f_A(\phi_i, T) and f_B(\phi_i, T) respectively. Specifically we put:

\[ f_A(\phi_i, T) = W_A \int \left( p - 1 \right) dp - \beta_i(T) \, dp, \]
\[ f_B(\phi_i, T) = W_B \int \left( p - 1 \right) dp - \beta_i(T) \, dp, \]

where here W_A, W_B are constants, and T the temperature is a parameter in this isothermal situation. We note that because \(-\frac{1}{\varepsilon} < T < T_m\), then \(-\frac{1}{\varepsilon} < \beta_i(T) < 0 < \beta_i(T) < 1\). The total energy density f(\phi, c_i, T) of the solution is:

\[ f(\phi, c_i, T) = c_i f_B(\phi, T) + \left( 1 - c_i \right) f_A(\phi, T) + \frac{kT}{v_m} \left[ \varepsilon \phi \log \left( 1 - c_i \right) \right], \]

where ε is Boltzmann constant and v_m is the molar volume. The first two terms correspond to the contributions to the energy density, due to the individual molar Gibbs free energies of the two species and the last term is due to the decrease in energy associated with the mixing of the two constituents, and our assumption that it is an ideal solution.

In a similar way to Kobayashi we define the free energy functional by:

\[ \mathcal{F}(\phi, c, T) = \int_0^2 \left( \frac{\nabla \phi \nabla \phi}{2} + f(\phi, c, T) \right) \, d\Omega. \]

In order to derive a kinetic model we make the assumption that the system evolves in time so that its total free energy decreases monotonically. This may be done by assuming the rate of change of c and φ vary according to the constrained gradient of \(\mathcal{F}(\phi, c, T)\):

\[ \dot{\phi} = -\eta \nabla \mathcal{F}(\phi), \]

where \(\eta = (\phi, c, T)\). Fife, [13] discusses such how constrained gradients may defined in a more rigorous mathematical setting. The only constraint we require here is that both species are conserved, i.e. \(\frac{\partial}{\partial t} \int_0^\Omega \mathcal{F} = 0\). We chose the constrained gradient such that:

\[ \frac{\partial \phi}{\partial t} = -\kappa_1 \frac{\partial \mathcal{F}}{\partial \phi} \]
\[ \frac{\partial c}{\partial t} = \kappa_2 \left[ \phi \left( \frac{\partial f_A}{\partial \phi} + (1 - c) \frac{\partial f_B}{\partial \phi} \right) \right], \]

where \(\kappa_1\) and \(\kappa_2\) are constants. The boundary conditions are:

\[ \frac{\partial \phi}{\partial n} = \frac{\partial c}{\partial n} = 0, \]

References


OF THE RADIATION TRANSFER PROBLEM IN AN ATMOSPHERE
WITH THE INHOMOGENEOUS SPREADING SURFACE

SULTANGAZIN U.M.
Institute of Mathematics & mechnics
Academy of Sciences of Kazakh SSR
Pushkin str.125, Alma-Ata, USSR

AND

MULDASHOV T.Z.
Institute of Mathematics & Mechanics
Academy of Sciences of Kazakh SSR
Pushkin str.125, Alma-Ata, USSR

Abstract-A method is presented for solving the multidimensional equation of radiative transfer in a scattering plane-parallel atmosphere with inhomogeneous spreading surface. The method, based on the spherical harmonics expansion, can be used to compute models with an arbitrary large optical thickness and any scattering phase function.

The correction problem of the distortion of the representation of the earth's surface arises in connection with the investigations of natural resources of the Earth from the space. The main stage here is the numerical solving of the following problem of radiative transfer in three-dimensional plane slab:

\[ \left( \Delta + \sigma(z) \right) I = S I, \]
\[ I|_{z=0} = \pi \int_{\Omega} \Delta \cdot d\omega', \omega \in \Omega_+, \]
\[ I|_{z=H} = 0, \quad \omega \in \Omega_. \]

where \( S = \frac{\sigma(\omega)}{4\pi} I(\hat{r}, \hat{\omega}) d\hat{\omega}. \)

\( I(\hat{r}, \hat{\omega}) \) is the intensity density at the point \( (x, y, z) \) along the unit vector \( \hat{\omega} = (\cos \theta, \sin \theta, \mu) \), \( \Omega \) is the surface of the unit sphere, \( \Omega_+ = \{ \hat{\omega} : \hat{\omega} \cdot \hat{n} \geq 0 \}, \)
\( \Omega_- = \{ \hat{\omega} : \hat{\omega} \cdot \hat{n} < 0 \}, \sigma(z) \) are the extinction and scattering coefficients respectively, \( g(z, \mu, \hat{\omega}, \omega) \) is the scattering phase function.

Boundary condition at the \( z=0 \) defines the illumination of the top of an atmosphere by a unidirectional beam of monochromatic radiation of strength \( \pi F_0 \). Condition at the \( z=H \) is the Lambert's reflection law, where \( 0 \leq g(x, y) \leq 1 \) is the reflection coefficient of the spreading surface.

By the method described I.V. Mishin and T.A. Sushkevich [1] the solution of the problem (1) can be reduced to the solving of the problems (2) and (3).

\[ \left( \mu \frac{\partial I}{\partial z} + \sigma(z) I = S I, \right. \]
\[ I|_{z=0} = \pi \int_{\Omega} \Delta \cdot d\omega', \omega \in \Omega_+, \]
\[ I|_{z=H} = 0, \quad \omega \in \Omega_. \]

The numerical solution of the problem (2) considered in our work [2]. And now we discuss the use of the spherical harmonics method for solving (3) when \( g \neq 0 \).

In this case, \( \Psi(z, \mu, \phi, \rho) \) can be expressed in a Fourier series as

\[ \Psi = \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} a_n^m(z, \mu) e^{im\phi} \cos nm \rho. \]

Phase function allowing following expansion:

\[ g(z, \mu, \phi, \rho) = \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} a_n^m(z, \mu) e^{im\phi} \cos nm \rho. \]

where \( \gamma^m = \sum_{k} g_k(z) \lambda_k^m(\mu) \lambda_k^m(\rho), \lambda_k^m(\mu) \) is the normalized associated Legendre polynomials.

By substituting (4) and (5) in (3) we obtain

\[ \left[ \mu \frac{\partial \Psi}{\partial z} + \sigma \Psi - \mathbf{C}(z) \Psi = \mathbf{S} \Psi, \right. \]
\[ \Psi|_{z=0} = 0, \quad \Psi|_{z=H} = 1, \quad \Psi|_{z=H} = 0, \quad m \geq 1, |\mu| < 1; \]

where \( \mathbf{C}(z) \) is the complex function, i.e. \( \Psi = \Psi_0 + i \Psi_1 \). By analyzing (6) we can to show that \( \Psi_{m=1} = 0 \) in \( \Psi_{m=0}, k=0,1, \ldots, [H/2]. \)

By eliminating those components from (6) and defining the new real function \( \Psi'_{m}(z, \mu, \rho) \) we find

\[ \left[ \mu \frac{\partial \Psi'}{\partial z} + \mathbf{C}(z) \Psi' = \mathbf{S} \Psi', \right. \]
\[ \Psi'|_{z=0} = 0, \quad \Psi'|_{z=H} = 1, \quad \Psi'|_{z=H} = 0, \quad m \geq 1, |\mu| < 1; \]

Expanding \( \Psi'_{m}(z, \mu, \rho) \) in Legendre polynomials

\[ \Psi'_{m} = \sum_{k}^{\infty} \frac{2k+1}{2} \lambda_k(z, \mu) \lambda_k(\rho). \]

Substituting this expansion in (7) we obtain the system of ordinary differential equations

\[ \mathbf{D} \Psi' = \mathbf{C}(z) \Psi', \quad \Psi'|_{z=0} = 0, \quad \Psi'|_{z=H} = 1, \quad \Psi'|_{z=H} = 0, \quad m \geq 1, \mu < 1; \]

where \( \mathbf{D} \) is the block-triadiagonal matrix with zero blocks in the main diagonal and blocks \( D_{m,m+1} = -D_{m+1,m}, \) are the rectangular matrices \( (N_m-1) \times (N_{m+1}-m) \).

For approximation of the boundary conditions in (7) we use Marshak's conditions

\[ G_1^0(z, \mu) = 0, \quad G_2^0(z, \mu) = 1, \quad G_2^{m+1}(z, \mu) = 0, \quad m \geq 1, \]

where \( G_1^0, G_2^0 \) are defined in [2].

For inhomogeneous in the vertical direction on atmospheric models, the usual approximati- on is to divide the atmosphere into several homogeneous layers. In each layer \( [z_{k-1}, z_k] \) matrix \( \mathbf{C}(z) \) is the constant and equal \( \mathbf{C}_z \).

Integrating (9) over \( z \) in each layer, we get

\[ \psi(z_{k-1}) + \exp(\Delta z_k) \psi(z_k) = 0, \]

where \( \psi(z_k) \) is the solution in each layer.
Integrating (14) without SW in right hand side we obtain \( W(z,\mu,\varphi) \), which is the approximation of \( W \) in the "single scattering". In order to evaluate \( W \) when \( \mu<0 \) it is necessary to compute the integrals
\[
\Delta z \int \exp(t/\mu) \sin bt \cos t \, dt = \frac{\sin b\mu}{\cos \mu},
\]
which can be obtained by multiplying of the system (9) at \( \exp(t/\mu) \sin bt \), by integrating over \( z \) and performing respectively matrix transformations. Function \( W \) is the exact value of the error of the spherical harmonics approximation of the function \( \Psi \), which is the component of the solution of the problem (3) and described "single reflection" from the spreading surface, i.e. \( \Psi \) has following form:
\[
\Psi = \Psi_{+} + \chi,
\]
where
\[
\chi(z,\mu,\varphi) = \exp[\i\pi \mu \varphi / \mu] \cos t - \exp[\i\pi \mu \varphi / \mu] \sin t.
\]

Thus, the describing smoothing procedure will make possible to use the spherical harmonics for solution the problem (3) with high efficiency.

Results of the solving of the problem (3) for the Elterman's atmospheric model [4] for \( \lambda=0.75 \) km are presented in figure. Here plotted function \( \chi \) at the top of atmosphere, which usually presented in following form:
\[
\chi(0,\mu,\varphi, p) = \exp[i\pi \mu \varphi / \mu] \cos t - \exp[i\pi \mu \varphi / \mu] \sin t.
\]

Integrating (14) without SW in right hand side we obtain \( W(z,\mu,\varphi) \), which is the approximation of \( W \) in the "single scattering". In order to evaluate \( W \) when \( \mu<0 \) it is necessary to compute the integrals
\[
\Delta z \int \exp(t/\mu) \sin bt \cos t \, dt = \frac{\sin b\mu}{\cos \mu},
\]
which can be obtained by multiplying of the system (9) at \( \exp(t/\mu) \sin bt \), by integrating over \( z \) and performing respectively matrix transformations. Function \( W \) is the exact value of the error of the spherical harmonics approximation of the function \( \Psi \), which is the component of the solution of the problem (3) and described "single reflection" from the spreading surface, i.e. \( \Psi \) has following form:
\[
\Psi = \Psi_{+} + \chi,
\]
where
\[
\chi(z,\mu,\varphi) = \exp[\i\pi \mu \varphi / \mu] \cos t - \exp[\i\pi \mu \varphi / \mu] \sin t.
\]

Thus, the describing smoothing procedure will make possible to use the spherical harmonics for solution the problem (3) with high efficiency.

Results of the solving of the problem (3) for the Elterman's atmospheric model [4] for \( \lambda=0.75 \) km are presented in figure. Here plotted function \( \chi \) at the top of atmosphere, which usually presented in following form:
\[
\chi(0,\mu,\varphi, p) = \exp[i\pi \mu \varphi / \mu] \cos t - \exp[i\pi \mu \varphi / \mu] \sin t.
\]

Integrating (14) without SW in right hand side we obtain \( W(z,\mu,\varphi) \), which is the approximation of \( W \) in the "single scattering". In order to evaluate \( W \) when \( \mu<0 \) it is necessary to compute the integrals
\[
\Delta z \int \exp(t/\mu) \sin bt \cos t \, dt = \frac{\sin b\mu}{\cos \mu},
\]
which can be obtained by multiplying of the system (9) at \( \exp(t/\mu) \sin bt \), by integrating over \( z \) and performing respectively matrix transformations. Function \( W \) is the exact value of the error of the spherical harmonics approximation of the function \( \Psi \), which is the component of the solution of the problem (3) and described "single reflection" from the spreading surface, i.e. \( \Psi \) has following form:
\[
\Psi = \Psi_{+} + \chi,
\]
where
\[
\chi(z,\mu,\varphi) = \exp[\i\pi \mu \varphi / \mu] \cos t - \exp[\i\pi \mu \varphi / \mu] \sin t.
\]
The synthesis of simplest algorithms for numerical solution of boundary-value problems with two or three space variables on the base of algorithms for corresponding single-dimensional problems is generally connected only with overcoming technical difficulties. But the quality of such simplest algorithms often proves to be low. For instance, the solution symmetry disturbance or the approximation loss can occur when the vector field is turned. The synthesis of more perfect algorithms for multidimensional problems is connected with overcoming serious mathematical difficulties.

Let's consider the system of equations for unsteady inviscous compressible fluid flows with three spatial variables.

\[
\begin{align*}
\frac{\partial \rho}{\partial t} + \text{div} \, p \mathbf{u} &= 0, \\
\frac{\partial e}{\partial t} + \text{div} \, p \mathbf{u} &= 0, \\
\frac{\partial \mathbf{u}}{\partial t} + \text{grad} \, p + \mathbf{rot} \, \mathbf{u} &= 0.
\end{align*}
\]

(1)

\[\rho - \text{density}, \quad e - \text{pressure}, \quad \mathbf{u} - \text{velocity of gas}, \quad \mathbf{e} - (\rho (e + 0.5|\mathbf{u}|^2), \quad e - \text{internal energy.}
\]

Equation of state for \(p, \rho\) and \(e\) must be added to the system (1).

**THEOREM.** Difference scheme of Godunov's type with strong symmetry conservation property reproduces the property of the initial boundary-value problem solution to transform in the solution of the same problem under the effect of shift or rotation groups. At the same time the same algorithm must be adaptable to any group irrespective of whether the group is the shift or rotation group and also irrespective of the shift direction and location of rotation centres or axes. i.e. whatever shift or rotation group of the initial-value-boundary problem withstands the numerical solution will transform into itself under the effect of the group discrete analogue.

Here we give the strict mathematical definition of the symmetry conservation property, distinguishing between the strong and weak symmetry conservation.

The algorithm of boundary-value problem numerical solution consists of the definition of the set serving as a basis for the approximate solution (for the counting region) and for finding the solution. We shall consider the counting region to be an image of a four-dimensional parallelepiped

\[\mathcal{E} = \{ (x^1, x^2, x^3, x^4) : 0 \leq x^1 \leq L, 0 \leq x^2 \leq L, 0 \leq x^3 \leq L, 0 \leq x^4 \leq K \} \]

\[(L, K, K, K) \text{ are positive integers.}
\]

The piecewise-smooth one-to-one intra \(\mathcal{E}\) mapping \(Y \rightarrow \mathbb{R}^4\) has the following form:

\[x = (x^1, x^2, x^3, x^4), y = (y^1, y^2, y^3, y^4), z = (z^1, z^2, z^3, z^4), t = (t^1, t^2, t^3, t^4)
\]

Here \(x, y, z\) are the cartesian coordinates in the space \(\mathbb{R}^3\), \(t\) is the time. We shall write \(t_k\) instead of \((k_1, k_2, k_3, k_4)\).

We consider the mapping \(Y\) to be linear over each line segment

\[k_1 \leq x_1 \leq k_2, \quad k_3 \leq x_3 \leq k_4, \quad k = 1, \ldots, K.
\]

We use \(\mathcal{E} = \{ (x^1, x^2, x^3, x^4) : 1 \leq x_1 \leq m, \quad m - 1 \leq x_2 \leq m, \quad n - 1 \leq x_3 \leq n, \quad x_4 = k \}
\]

The approximate solution is a function defined on the set \(Y (\mathcal{E})\) and constant on every set \(Y (\mathcal{E} = \{(x^1, x^2, x^3, x^4) : 1 \leq x_1 \leq m, \quad 1 \leq x_2 \leq n, \quad 0 \leq k \leq K\})\). We define the value of the approximate solution on this set as \(f_{m,n,k}\).

**Definition 1.** The boundary-value problem is called two-dimensional (one-dimensional) in the following case. First, there is such a \(\alpha, \beta, \gamma\) system of rectangular cartesian, or cylindrical, or spherical coordinates at the space \(\mathbb{R}^3\) that the problem solution can be defined as a function of the variables \(\alpha, \beta, \gamma\) the components that correspond to the unit vectors of the variables \(\beta, \gamma\) are to be zero.

Later we shall consider only two- or one-dimensional problems, for which the variables \(\alpha, \beta, \gamma\) are not angular. It means that the boundary-value problem withstands the effect of the shift or rotation group, and the extent group.

**Definition 2.** We shall say the grid construction method has the symmetry conservation property, if any two- or one-dimensional problem in the coordinate system \(\alpha, \beta, \gamma\) appropriate for a given problem satisfies the equalities

\[Y (1, m, n, t_k) = (\alpha_1, m, \beta_1, n, \gamma_k) 
\]

- for two-dimensional problems and

\[Y (1, m, n, t_k) = (\alpha_1, m, \beta_1, n, t_k) 
\]

- for one-dimensional problems. Here

\[\alpha_1 < \alpha_2 < \ldots < \alpha_L, \quad \beta_0 < \beta_1 < \ldots < \beta_M, \quad \gamma_0 < \gamma_1 < \ldots < \gamma_N
\]

**Definition 3.** The numerical solution algorithm for a class of boundary-value problems with the three space variables has the weak symmetry conservation property, if the grid construction method has the symmetry conservation property and the numerical solution of any two-dimensional problem of the class considered, based on this algorithm satisfi-
es the condition
\[ f_{1m1k} = \ldots = f_{1mNk} = f_{1mk}; \]
\[ 1 \leq l \leq L, 1 \leq m \leq M, 0 \leq k \leq K; \]
and the numerical solution of any one-dimensional problem of the class considered satisfies the condition
\[ f_{111k} = \ldots = f_{11Mk} = f_{112k} = \ldots = f_{1M2k} = \ldots = f_{1MNk} = \]
\[ = f_{1k}; \]
\[ 1 \leq l \leq L, 0 \leq k \leq K. \]

If \( f_{1m1k} \) contains vectors, then we consider the vector equality as an equality of their components obtained by their decomposition into unit vectors of the coordinate system \( \alpha \beta \gamma \).

Definition 4. The numerical solution algorithm for a class of boundary-value problems with the three space variables has the strong symmetry conservation property, if it has the weak symmetry conservation property and of any two-dimensional problems of the class considered the numerical solution \( f_{1m1k} \) does not depend on the number \( N \) and the set \( \gamma_0, \gamma_1, \ldots, \gamma_N \), and of any single-dimensional problem of the class considered the numerical solution \( f_{1k} \) doesn't depend on the numbers \( M \) and \( N \) and the sets \( \beta_0, \beta_1, \ldots, \beta_M \) and \( \gamma_0, \gamma_1, \ldots, \gamma_N \).

An illustration will make the difference between weak and strong symmetry conservations clear. Let the decision of boundary-value problem depend on the \( r \) coordinate only in the \((z,r,\phi)\) cylindrical coordinate system. The different grids for this problem solution are presented in Fig.1,2 in the \( z=const \) plane. These grids have the same cell number in the \( r \) direction and the different cell number in \( \phi \) direction. If the algorithm has the weak symmetry conservation property, then the solution will be identical both for the grid shown in Fig.1 and for the grid shown in Fig.2 for all the cell layers satisfying the fixed \( \phi \). However, the solution \( F_1(\; r \; ) \) obtained using the first grid, can be different from the solution \( F_2(\; r \; ) \) obtained using the second grid. If the algorithm has the strong symmetry conservation property, \( F_1(\; r \; ) \) fits \( F_2(\; r \; ) \).

![Fig.1](image1.png)

![Fig.2](image2.png)

The important property of mathematical physical problems is the independence from the coordinate system, in which they are given since the problem itself does not depend on any coordinate system. In other words, the equations of mathematical physics are written in the invariant vector (or tensor) form. It is the numerical algorithm that is reproduction of this property in analysis. We apply the term "invariance" to the calculation results independence on choice of the \( x,y,z \) coordinate system in the space \( \mathbb{R}^3 \) and parameterization (the mapping \( \psi \)). Our difference scheme has the invariance property. The symmetry conservation and invariance properties are important in the numerical solution of complex geometry problems, where the solution can have various kinds of symmetry in various parts of counting region and when it is not clear what kind of coordinate system shall we choose to carry out the analysis. The strong symmetry conservation property is also correspondence with the plane points constructed. By this way the surface is parametrically represented.

The multi-dimensional analogue of the Fa-ber - Schauder basis for the space of functions was created for the substantiation of the surface construction.

Three problems are solved for construction of the difference scheme. 1.Construction of curvilinear cells. 2.Obtaining of invariance of equations of motion close to two- or one-dimensional ones. If the numerical algorithm has the strong symmetry conservation property, then we can solve two- and one-dimensional problems using the programs intended for three dimensional calculations.

Three problems are solved for construction of curvilinear surfaces is proposed. The novelty of the method suggested is that, the countable set of points situated on the surface is generated to begin with. Then it is proved that, there is the Lipshitz mapping of the parametric plane rectangle into a space \( \mathbb{R}^3 \) where the binary-rational points of the rectangle are in one-to-one correspondence with the plane points constructed. By this way the surface is parametrically represented.

The multi-dimensional analogue of the Fa-ber - Schauder basis for the space of functions was created for the substantiation of the surface construction.

Three surface families covering the domain so that one and only one surface of each of the families goes through the each point of the domain generate three scalar equations of motion. Each of the equations is generated by one of the surface families. These equations are including only the \( v=const \) components obtained by its decomposition into duality that consist of the vectors going normal to or in the main directions of surfaces. All the geometric entities involved in equations do not depend on ways of specifying or parameterizing surfaces. They are defined by invariants of the first and second quadratic forms. The differentiation is included into the obtained equations only as a divergence of some vector fields.

Proof of the theorem is given in (1).