REPORT RE-799

HIGHLY ACCURATE PREDICTION OF UNSTEADY VISCOUS FLOWS

APRIL 1992

GRUMMAN®

CORPORATE RESEARCH CENTER

REPORT RE-799

HIGHLY ACCURATE PREDICTION OF UNSTEADY VISCOUS FLOWS

APRIL 1992

by

F. Marconi
Grumman Aerospace Corporation
Bethpage, New York 11714

and

Gino Moretti
G. M. A. F., Inc.
Freeport, New York 11520

Final Report on Contract F49620-91-C-0015
for Period 1 January 1991 - 1 January 1992

Prepared for

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
Bolling Air Force Base
Washington, D. C. 20332

Approved by: Richard D Iasi

Richard De Iasi, Director
Corporate Research Center

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
April 1992

3. REPORT TYPE AND DATES COVERED
Final Report 1/1/91-1/1/92

4. TITLE AND SUBTITLE
HIGHLY ACCURATE PREDICTION OF UNSTEADY VISCOUS FLOWS

5. FUNDING NUMBERS
2307/AC
PE 61102F
F49620
91-C-0015

6. AUTHOR(S)
F. Marconi and G. Moretti

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Grumman Aerospace Corporation
Bethpage, NY 11714

8. PERFORMING ORGANIZATION REPORT NUMBER
RE-

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research
Bolling AFB, Washington, D.C. 20332
NA

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
F49620-91-C-0015

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
UNLIMITED

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report describes a detailed investigation of the effects of computational accuracy on the prediction of shock wave/boundary layer interaction. In particular, the result of inaccuracies in the computation of the shock and the flow in its vicinity is studied. A new computational procedure computes all shocks as discontinuities while including all viscous effects. This scheme is used as a standard against which the accuracy of widely used shock capturing schemes is measured. The effect of the numerical error generated by spreading a shock over a few mesh intervals (instead of a few mean free paths) is evaluated with regard to shock/boundary layer interaction. We consider the spreading error, as well as the error produced by reducing the formal accuracy of these schemes near shocks (in order to eliminate wiggles). In this report, we present the computational scheme and results for a number of flow configurations.

14. SUBJECT TERMS
Computational Fluid Dynamics, Shock Wave/Boundary Layer Iteration

15. NUMBER OF PAGES
43

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT

ABSTRACT

This report describes a detailed investigation of the effects of computational accuracy on the prediction of shock wave/boundary layer interaction. In particular, the result of inaccuracies in the computation of the shock and the flow in its vicinity is studied. A new computational procedure computes all shocks as discontinuities while including all viscous effects. This scheme is used as a standard against which the accuracy of widely used shock capturing schemes is measured. The effect of the numerical error generated by spreading a shock over a few mesh intervals (instead of a few mean free paths) is evaluated with regard to shock/boundary layer interaction. We consider the spreading error, as well as the error produced by reducing the formal accuracy of these schemes near shocks (in order to eliminate wiggles). In this report, we present the computational scheme and results for a number of flow configurations.

CONTENTS

LIST OF ILLUSTRATIONS

I. INTRODUCTION

Over the past two decades, computational fluid dynamics (CFD) has matured to the point that it contributes significantly to the understanding of fluid flows. Now our main concern is the reliability of the predictions of CFD, and therefore accuracy of CFD algorithms should be the main driver of our research. While advances in computer technology may enable CFD researchers to resolve essentially smooth portions of complex flow fields, special considerations will have to be paid to "singular regions" to ensure accuracy. Adaptive gridding/grid clustering will help resolve geometric singularities (wing leading edge, for example), boundary layers, vortex sheets and vortex centers, but to resolve shock waves may be too much to ask of any grid-adapting scheme. The shock wave is, in fact, a discontinuity under the assumption of continuum flow (its thickness being on the order of a few mean free paths) while boundary layers, contact sheets and vortex centers spread as $\sqrt{\mathrm{Re}_x}$ with x measured from their origin, in laminar flow (turbulent mixing is faster). We offer the alternative of fitting the shocks in the flow field while resolving all other "singularities" with grid adaptation. In this way very high degrees of accuracy can be achieved for the shock system and the viscous regions as well (boundary layer, contact sheet and vortices).

With this approach, we feel that the accurate/reliable prediction of 3-D unsteady viscous flows is feasible, if not yet with current supercomputers, then using the next generation of multiple processor machines. The equations to be solved are the Reynolds-averaged Navier-Stokes equations with an appropriate model for the turbulent shear stresses. The computational procedure used in smooth portions of the flow will assure consistency between the numerical and physical domains of dependence. Shock waves will be fit and the exact Rankine-Hugoniot jump conditions satisfied across them. Grid adaptation can be used to resolve geometric singularities, boundary layers, vortex sheets and vortex centers.

In the phase of our effort reported upon here, we were able to demonstrate the capability of our computational procedure by applying our scheme to a number of unit problems. The ability to combine the accuracy of fitting shock waves with a full Reynolds-averaged Navier-Stokes model was demonstrated for the first time. We start with a detailed description of the computational procedure developed during this phase of our work, followed by sample calculations and, finally, some concluding remarks and recommendations for future work.

II. DETAILS OF THE COMPUTATIONAL PROCEDURE

1. NON-DIMENSIONALIZATION

Non-dimensional physical quantities are defined as follows:

Let $\bar{p}, \bar{\rho}, \bar{a}, \bar{T}, \bar{S}$ be dimensional values of pressure, density, speed of sound, temperature and entropy, respectively, and let $\bar{\mathbf{q}}$ be the dimensional velocity vector. Let $\bar{c}_p, \bar{c}_v$ be the specific heats at constant pressure and constant volume, respectively, and

$$\mathcal{R} = \bar{c}_p - \bar{c}_v, \qquad \gamma = \bar{c}_p/\bar{c}_v, \qquad \delta = (\gamma - 1)/2, \qquad \bar{c}_p/\mathcal{R} = \gamma/2\delta \tag{1.1}$$

Assuming the values of pressure, density and temperature at infinity as reference values, we define non-dimensional pressures, densities and temperatures p, ρ and T as

$$p = \bar{p}/\bar{p}_\infty, \quad \rho = \bar{\rho}/\bar{\rho}_\infty, \quad T = \bar{T}/\bar{T}_\infty \tag{1.2}$$

Since

$$\bar{p} = \mathcal{R}\bar{\rho}\bar{T}, \quad \bar{p}_\infty = \mathcal{R}\bar{\rho}_\infty\bar{T}_\infty \tag{1.3}$$

the basic relation between p, ρ and T is written as:

$$p = \rho T \tag{1.4}$$

Also, $\mathcal{R}\bar{T}_\infty$ is assumed as the square of the reference velocity, $\bar{q}_{\text{ref}}$; therefore, the non-dimensional speed of sound, a, is

$$a = (\gamma T)^{1/2} \tag{1.5}$$

A suitable reference length, $\bar{x}_{\text{ref}}$, having been assumed, the reference time, $\bar{t}_{\text{ref}}$, is assumed as the ratio of the reference length and the reference velocity:

$$\bar{t}_{\text{ref}} = \bar{x}_{\text{ref}}/\bar{q}_{\text{ref}} \tag{1.6}$$

Finally, we assume the entropy at infinity equal to zero, and we non-dimensionalize the entropy by the formula:

$$S = \bar{S}/(\gamma\mathcal{R}) \tag{1.7}$$

The Mach number at infinity, M_∞, is

$$M_\infty = \bar{q}_\infty/\bar{a}_\infty \tag{1.8}$$

Additional definitions are needed for viscous flows. Let $\bar{\Delta}$ be the divergence of $\bar{\mathbf{q}}$:

$$\bar{\Delta} = \nabla.\bar{\mathbf{q}} \tag{1.9}$$

$\bar{\mathbf{s}}$ the stress tensor minus the diagonal terms containing pressure, $\bar{\mathbf{Q}}$ the heat-flux vector and $\bar{\Phi}$ the dissipation function. The stress tensor (a rather complicated function of space derivatives of the velocity components) depends linearly on the viscosity, $\bar{\mu}$:

$$\bar{\mathbf{s}} = 2\bar{\mu}[\text{def } \bar{\mathbf{q}} - \frac{1}{3}\bar{\Delta}\mathbf{D}] \tag{1.10}$$

where **D** is the unit diagonal tensor and

$$\text{def } \bar{\mathbf{q}} = \frac{1}{2}[\nabla \bar{\mathbf{q}} + (\nabla \bar{\mathbf{q}})^T] \tag{1.11}$$

(the apex T denotes the transpose of the tensor). The heat-flux vector is proportional to the gradient of the temperature:

$$\bar{\mathbf{Q}} = -\bar{k}\nabla \bar{T} \tag{1.12}$$

The coefficient, $\bar{k}$, is the thermal conductivity. We also define a "dissipation", $\bar{\Phi}$, as

$$\bar{\Phi} = 2\bar{\mu}(\text{def } \bar{\mathbf{q}} \cdot \text{def } \bar{\mathbf{q}} - \frac{1}{3}\bar{\Delta}^2) \tag{1.13}$$

We choose reference values for $\bar{\mu}$ and $\bar{k}$, for example $\bar{\mu}_\infty$ and $\bar{k}_\infty$ and define a Reynolds number,

$$\mathcal{R}_e = \frac{\bar{q}_\infty \bar{\rho}_\infty \bar{x}_{\text{ref}}}{\bar{\mu}_\infty} \tag{1.14}$$

and a Prandtl number,

$$\mathcal{P}_r = \frac{\bar{\mu}_\infty \bar{c}_p}{\bar{k}_\infty} \tag{1.15}$$

The continuity equation, in dimensional form, is:

$$\frac{D\bar{\rho}}{D\bar{t}} + \bar{\rho}\nabla \cdot \bar{\mathbf{q}} = 0 \tag{1.16}$$

The momentum equation, in dimensional form, is:

$$\frac{D\bar{\mathbf{q}}}{D\bar{t}} + \frac{1}{\bar{\rho}}\nabla \bar{p} = \frac{1}{\bar{\rho}}\nabla \cdot \bar{\mathbf{s}} \tag{1.17}$$

and the Lagrangian increment in entropy is:

$$\frac{D\bar{S}}{D\bar{t}} = \frac{1}{\bar{\rho}\bar{T}}\,[\bar{\Phi} - \nabla \cdot \bar{\mathbf{Q}}] \tag{1.18}$$

The above definitions allow the three equations of motion to be recast in non-dimensional form. The continuity equation remains formally the same, without the bars:

$$\frac{D\rho}{Dt} + \rho\nabla \cdot \mathbf{q} = 0 \tag{1.19}$$

The momentum equation becomes:

$$\frac{D\mathbf{q}}{Dt} + \frac{1}{\rho}\nabla p = \mathbf{V}_m \tag{1.20}$$

where

$$\mathbf{V}_m = \frac{M_\infty\sqrt{\gamma}}{\mathcal{R}_e}\,\frac{1}{\rho}\,\nabla \cdot \mathbf{s} \tag{1.21}$$

(s is formally the same as $\bar{s}$ without the bars). The entropy equation becomes

$$\frac{DS}{Dt} = V_s \tag{1.22}$$

with

$$V_s = \frac{M_\infty}{\sqrt{\gamma}\mathcal{R}_e} \, \frac{1}{p} \, [\Phi - \frac{\gamma}{\gamma - 1} \, \frac{1}{\mathcal{P}_r} \, \nabla \cdot \mathbf{Q}] \tag{1.23}$$

where the non-dimensional dissipation and the non-dimensional heat-flux are formally the same as $\bar{\Phi}$ and $\bar{\mathbf{Q}}$, without the bars. We will consider the viscosity and the heat conduction coefficients as functions of the temperature:

$$\mu = T^{3/2} \frac{1 + \bar{T}_0/\bar{T}_\infty}{T + \bar{T}_0/\bar{T}_\infty} \tag{1.24}$$

(where $\bar{T}_0$ is an appropriate constant; for example 110° Kelvin), and

$$k = T^{3/4} \tag{1.25}$$

2. UPWIND ("λ") REFORMULATION OF THE EQUATIONS

The equations of motion (1.19), (1.20) and (1.22) can now be reformulated to emphasize the role of the propagation of signals in the convective terms and the diffusive role of the viscous terms. The new basic variables are a, S and $\mathbf{q}$. From the definition of entropy in a perfect gas,

$$S = \frac{1}{2\gamma\delta}[\ln p - \gamma \ln \rho] = \frac{1}{2\gamma\delta} \ln \frac{a^2}{\gamma} - \frac{1}{\gamma} \ln \rho \tag{2.1}$$

it follows that, for any derivative, here denoted by a prime,

$$S' = \frac{1}{2\gamma\delta}[\frac{p'}{p} - \gamma \frac{\rho'}{\rho}] \tag{2.2}$$

$$S' = \frac{1}{\gamma\delta} \frac{a'}{a} - \frac{1}{\gamma} \frac{\rho'}{\rho} \tag{2.3}$$

$$S' = \frac{1}{\delta} \frac{a'}{a} - \frac{1}{\gamma} \frac{p'}{p} \tag{2.4}$$

Using (2.3), (1.19) becomes

$$\frac{1}{\delta} \frac{Da}{Dt} + a \, \nabla \cdot \mathbf{q} = \gamma a \frac{DS}{Dt} \tag{2.5}$$

Using (2.4), (1.4) and (1.5), (1.20) becomes:

$$\frac{D\mathbf{q}}{Dt} + \frac{a}{\delta} \nabla a - a^2 \nabla S = \mathbf{V}_m \tag{2.6}$$

To make the computational technique for viscous flows compatible with the λ-formulation for inviscid flows, we split S in (2.5) into two terms:

$$S = (\frac{1}{\gamma} + \frac{2\delta}{\gamma})S \tag{2.7}$$

recasting (2.5) into the form:

$$\frac{1}{\delta}\frac{Da}{Dt} + a\nabla\cdot\mathbf{q} - a\frac{DS}{Dt} = 2\delta a V_s \tag{2.8}$$

At this point, (2.8), (2.6) and (1.22) can be treated by the general procedure exposed in [1], with the addition of the source terms, $2\delta a V_s$, $\mathbf{V}_m$, and V_s, respectively.

The flow is two-dimensional, and the computational grid is orthogonal. In the initial phase of the computation, the rigid walls and the computational frame move from right to left, accelerating from a state of rest to the constant velocity,

$$q_\infty = M_\infty\sqrt{\gamma} \tag{2.9}$$

. The equations of motion are recast for the general case of an accelerating frame. This changes only the momentum equation (2.6), to which a term, $q_{\infty t}\mathbf{I}$ must be added in the right-hand side, with $q_{\infty t}$ being the acceleration of the frame and $\mathbf{I}$ the unit vector of the x-axis.

We rewrite the equations of motion after splitting the Lagrangian derivatives into local derivatives and convective terms:

$$\frac{1}{\delta}\frac{\partial a}{\partial t} + \mathbf{q}\cdot\frac{\nabla a}{\delta} + a\nabla\cdot\mathbf{q} - a(\frac{\partial S}{\partial t} + \mathbf{q}\cdot\nabla S) = 2\delta a V_s \tag{2.10}$$

$$\frac{\partial\mathbf{q}}{\partial t} + (\mathbf{q}\cdot\nabla)\mathbf{q} + \frac{a}{\delta}\nabla a - a^2\nabla S = q_{\infty t}\mathbf{I} + \mathbf{V}_m \tag{2.11}$$

$$\frac{\partial S}{\partial t} + \mathbf{q}\cdot\nabla S = V_s \tag{2.12}$$

These equations are the same as equations (1.2) in [1], with the addition of the source terms in the right-hand sides.

We proceed as explained in Section 6 of [1]. Four equations are obtained containing two unit vectors, orthogonal to each other, $\mathbf{i}$ and $\mathbf{j}$, that we choose parallel to the coordinate lines at each point (these vectors are a particular case of the unit vectors $\mathbf{n}$ and τ of [1]). Using indices to denote partial derivatives and calling u and v the components of $\mathbf{q}$ along the coordinate lines, the equations (similar to (2.1) of [1], but with right-hand sides), are:

$$\begin{aligned}
&a_t/\delta + u_t - aS_t + (\mathbf{q}+a\mathbf{i})\cdot[\nabla(a/\delta+u) - a\nabla S] + a\mathbf{j}\cdot\nabla v - \beta v + F = 2\delta a V_s + q_{\infty t}\mathbf{I}\cdot\mathbf{i} + \mathbf{V}_m\cdot\mathbf{i}\\
&a_t/\delta - u_t - aS_t + (\mathbf{q}-a\mathbf{i})\cdot[\nabla(a/\delta-u) - a\nabla S] + a\mathbf{j}\cdot\nabla v + \beta v + F = 2\delta a V_s - q_{\infty t}\mathbf{I}\cdot\mathbf{i} - \mathbf{V}_m\cdot\mathbf{i}\\
&a_t/\delta + v_t - aS_t + (\mathbf{q}+a\mathbf{j})\cdot[\nabla(a/\delta+v) - a\nabla S] + a\mathbf{i}\cdot\nabla u + \beta u + F = 2\delta a V_s + q_{\infty t}\mathbf{I}\cdot\mathbf{j} + \mathbf{V}_m\cdot\mathbf{j}\\
&a_t/\delta - v_t - aS_t + (\mathbf{q}-a\mathbf{j})\cdot[\nabla(a/\delta-v) - a\nabla S] + a\mathbf{i}\cdot\nabla u - \beta u + F = 2\delta a V_s - q_{\infty t}\mathbf{I}\cdot\mathbf{j} - \mathbf{V}_m\cdot\mathbf{j}
\end{aligned} \tag{2.13}$$

with

$$\alpha = \arccos(\mathbf{I}\cdot\mathbf{i}), \quad \mathbf{k} = \mathbf{i}\times\mathbf{j} \tag{2.14}$$

$$\beta = \mathbf{q}\cdot\nabla\alpha, \quad F = a\mathbf{k}\times\mathbf{q}\cdot\nabla\alpha \tag{2.15}$$

Adding the four equations (2.13), and subtracting (2.10) multiplied by 2, we obtain:

$$(2/\delta)a_t - 2aS_t + 2a\mathbf{q}\cdot\nabla S + U_1 + U_2 + U_3 + U_4 - (2/\delta)\mathbf{q}\cdot\nabla a = 4\delta a V_s \tag{2.16}$$

Subtracting the second of (2.13) from the first, we obtain:

$$2u_t + U_1 - U_2 = 2q_{\infty t}\cos\alpha + 2\mathbf{V}_m \cdot \mathbf{i} \tag{2.17}$$

Subtracting the fourth of (2.13) from the third, we obtain:

$$2v_t + U_3 - U_4 = 2q_{\infty t}\sin\alpha + 2\mathbf{V}_m \cdot \mathbf{j} \tag{2.18}$$

where

$$\begin{aligned}
U_1 &= (\mathbf{q} + a\mathbf{i}) \cdot [\nabla(a/\delta + u) - v\nabla\alpha - a\nabla S] \\
U_2 &= (\mathbf{q} - a\mathbf{i}) \cdot [\nabla(a/\delta - u) + v\nabla\alpha - a\nabla S] \\
U_3 &= (\mathbf{q} + a\mathbf{j}) \cdot [\nabla(a/\delta + v) + u\nabla\alpha - a\nabla S] \\
U_4 &= (\mathbf{q} - a\mathbf{j}) \cdot [\nabla(a/\delta - v) - u\nabla\alpha - a\nabla S]
\end{aligned} \tag{2.19}$$

These equations are the counterparts of (2.2), (2.3) and (2.4) in [1], with a different grouping of terms to eliminate the explicit appearance of F and β. Note that the first parenthesis in the right-hand sides of (2.19) contain vectors. For example, in the first equation,

$$\mathbf{q} + a\mathbf{i} = (u + a)\mathbf{i} + v\mathbf{j}$$

When expanding the expressions, the terms affected by $u + a$ must be approximated using differences taken from the side originating the $(u + a)$-signal. Such terms cannot be simplified with similar terms in the second equation, affected by the coefficient $u - a$. Contrariwise, the terms affected by the coefficient v can be simplified between the first two equations, because both have the same domain of dependence. The important simplifications that the above argument implies are shown in what follows, in the case of orthogonal coordinates.

The physical plane is defined by a complex variable $z = x + iy$ and is conformally mapped onto another plane, defined by a complex variable $\zeta = \xi + i\eta$. The grid in the ζ-plane is Cartesian; therefore its counterpart in the z-plane is orthogonal. It may be necessary to stretch the coordinate lines in the ζ-plane to obtain a better resolution in the z-plane. In this case, the rectangular grid in the ζ-plane is non-conformally related to another rectangular grid in a plane, defined by Cartesian coordinates X and Y, where X is a function of ξ alone and Y is a function of η alone; this does not impair the orthogonality of the final mesh. All gradients appearing in the left-hand sides of (2.16), (2.17) and (2.18) are conveniently computed in the ζ-plane. With

$$g = \frac{d\zeta}{dz}, \quad G = |g|, \quad \phi = \phi_1 + i\phi_2 = \frac{d\log g}{d\zeta} \tag{2.20}$$

we have:

$$\nabla\alpha = G(\alpha_\xi\mathbf{i} + \alpha_\eta\mathbf{j}) = -G(\phi_2\mathbf{i} + \phi_1\mathbf{j}) \tag{2.21}$$

and

$$\begin{aligned}
U_1 &= G(u + a)[(a/\delta + u)_\xi - v\alpha_\xi - aS_\xi] + Gv[(a/\delta + u)_\eta - v\alpha_\eta - aS_\eta] \\
U_2 &= G(u - a)[(a/\delta - u)_\xi + v\alpha_\xi - aS_\xi] + Gv[(a/\delta - u)_\eta + v\alpha_\eta - aS_\eta] \\
U_3 &= Gu[(a/\delta + v)_\xi + u\alpha_\xi - aS_\xi] + G(v + a)[(a/\delta + v)_\eta + u\alpha_\eta - aS_\eta] \\
U_4 &= Gu[(a/\delta - v)_\xi - u\alpha_\xi - aS_\xi] + G(v - a)[(a/\delta - v)_\eta - u\alpha_\eta - aS_\eta]
\end{aligned} \tag{2.22}$$

$$\mathbf{q}\cdot[\nabla(a/\delta)-a\nabla S]=G[(ua_\xi/\delta-auS_\xi)+(va_\eta/\delta-avS_\eta)] \tag{2.23}$$

The above mentioned simplifications occur in (2.16) because the second terms in U_1 and U_2 and the first terms in U_3 and U_4 cancel out the right-hand side of (2.23). On the other hand, the second terms of U_1 and U_2 reduce to $2Gv(u_\eta - v\alpha_\eta)$, and the first terms of U_3 and U_4 reduce to $2Gu(v_\xi + u\alpha_x i)$ when computing $U_1 - U_2$ and $U_3 - U_4$, respectively. For better clanty in coding, we define

$$\lambda_1^X = G(u+a),\ \ \lambda_2^X = G(u-a),\ \ \lambda_3^X = Gu$$

$$\lambda_1^Y = G(v+a),\ \ \lambda_2^Y = G(v-a),\ \ \lambda_3^Y = Gv \tag{2.24}$$

$$R_1^X = a/\delta+u,\ \ R_2^X = a/\delta-u,\ \ R_1^Y = a/\delta+v,\ \ R_2^Y = a/\delta-v$$

and

$$\begin{aligned} f_1^X &= -\lambda_1^X(R_{1\xi}^X - v\alpha_\xi - aS_\xi),\ \ f_2^X = -\lambda_2^X(R_{2\xi}^X + v\alpha_\xi - aS_\xi),\ \ f_3^X = -2\lambda_3^X(v_\xi + u\alpha_\xi)\\ f_1^Y &= -\lambda_1^Y(R_{1\eta}^Y + u\alpha_\eta - aS_\eta),\ \ f_2^Y = -\lambda_2^Y(R_{2\eta}^Y - u\alpha_\eta - aS_\eta),\ \ f_3^Y = -2\lambda_3^Y(u_\eta - v\alpha_\eta)\end{aligned} \tag{2.25}$$

and

$$f_4^X = -\lambda_3^X S_\xi,\ \ \ f_4^Y = -\lambda_3^Y S_\eta \tag{2.26}$$

The equations of motion are now:

$$S_t = f_4^X + f_4^Y + V_s \tag{2.27}$$

$$a_t = \frac{\delta}{2}[f_1^X + f_2^X + f_1^Y + f_2^Y + 2aS_t + 4\delta a V_s] \tag{2.28}$$

$$u_t = \frac{1}{2}[f_1^X - f_2^X + f_3^Y + 2q_{\infty t}\cos\ \alpha + 2\mathbf{V}_m\cdot\mathbf{i}] \tag{2.29}$$

$$v_t = \frac{1}{2}[f_1^Y - f_2^Y + f_3^X - 2q_{\infty t}\sin\ \alpha + 2\mathbf{V}_m\cdot\mathbf{j}] \tag{2.30}$$

We must now evaluate the viscous terms, V_s and $\mathbf{V}_m$. In orthogonal coordinates, generated by conformal mapping,

$$e_{11} = G(u_\xi + v\phi_2),\ \ e_{22} = G(v_\eta - u\phi_1),\ \ e_{12} = \frac{G}{2}(v_\xi + u_\eta - u\phi_2 + v\phi_1) \tag{2.31}$$

$$\Delta = e_{11} + e_{22} \tag{2.32}$$

The two components of $\nabla\cdot\mathbf{s}$ are

$$2G\{[\mu(e_{11} - \frac{1}{3}\Delta)]_\xi + (\mu e_{12})_\eta + 2\mu e_{12}\phi_2 + \mu(e_{22} - e_{11})\phi_1]\} \tag{2.33}$$

$$2G\{(\mu e_{12})_\xi + [\mu(e_{22} - \frac{1}{3}\Delta)]_\eta - 2\mu e_{12}\phi_1 + \mu(e_{11} - e_{22})\phi_2]\} \tag{2.34}$$

Using a classical notation, we define:

$$\tau_{11} = 2\mu(e_{11} - \frac{1}{3}\Delta),\ \ \tau_{12} = 2\mu e_{12},\ \ \tau_{22} = 2\mu(e_{22} - \frac{1}{3}\Delta) \tag{2.35}$$

Therefore, the two components of $\mathbf{V}_m$ can be written as:

$$\mathbf{V}_m \cdot \mathbf{i} = \frac{M_\infty\sqrt{\gamma}}{\mathcal{R}_e}\frac{G}{\rho}\left[(\tau_{11})_\xi + (\tau_{12})_\eta + 2\tau_{12}\phi_2 + (\tau_{22} - \tau_{11})\phi_1\right] \tag{2.36}$$

$$\mathbf{V}_m \cdot \mathbf{j} = \frac{M_\infty\sqrt{\gamma}}{\mathcal{R}_e}\frac{G}{\rho}\left[(\tau_{12})_\xi + (\tau_{22})_\eta - 2\tau_{12}\phi_1 + (\tau_{11} - \tau_{22})\phi_2\right] \tag{2.37}$$

From its definition (1.13),using (2.31) and (2.32), Φ can be written as:

$$\Phi = 4\mu[\frac{1}{3}(e_{11}^2 + e_{22}^2 - e_{11}e_{22}) + e_{12}^2] \tag{2.38}$$

Finally, the term $\nabla \cdot \mathbf{Q}$ is evaluated as

$$\nabla \cdot \mathbf{Q} = -\nabla \cdot (k\nabla T) = -G^2[(kT_\xi)_\xi + (kT_\eta)_\eta] \tag{2.39}$$

Let u_C and v_C be the Cartesian components of $\mathbf{q}$; thus,

$$\begin{aligned} u &= u_C \cos\ \alpha + v_C \sin\ \alpha \\ v &= -u_C \sin\ \alpha + v_C \cos\ \alpha \end{aligned} \tag{2.40}$$

and

$$\begin{aligned} u_C &= u\ \cos\ \alpha - v\ \sin\ \alpha \\ v_C &= u\ \sin\ \alpha + v\ \cos\ \alpha \end{aligned} \tag{2.41}$$

A) *Left computational boundary* – The flow is generally considered inviscid to the left of the computational boundary. The observer moves with the rigid walls; at the end of the accelerating phase, the flow at infinity appears as moving towards the observer at the speed defined by (2.9). If M_∞ is greater than 1, a is forced equal to $\sqrt{\gamma}$, u_C to q_∞, v_C and S to zero over the entire left computational boundary after each updating defined by (2.27), (2.28), (2.29) and (2.30), and the values of f_i^X, f_i^Y $(i = 1, 2, 3, 4)$ are all defined using (2.40) before such updatings.

If M_∞ is less than 1, v_C is still assumed to vanish along the left computational boundary (this is tantamount to directing the flow horizontally by means of an infinite cascade of vanes, and it is not a disturbing assumption). The values of all f_i^Y's are defined accordingly. The steady flow conditions to the left of the boundary are expressed by the constance of the stagnation speed of sound and by the vanishing of the entropy. Such conditions are sufficient to determine the values of a and u, separately, when combined with the value of R_2^X, that is obtained correctly using inside information only. Specifically,

$$a_0^2 = a^2 + \delta(u^2 + v^2) = a^2 + \delta u^2/\cos^2\alpha \tag{2.42}$$

and

$$R_2^X = a/\delta - u \tag{2.43}$$

From (2.42) we obtain:

$$aa_t/\delta + uu_t/\cos^2\alpha = 0 \tag{2.44}$$

and using (2.28) and (2.29) without viscous, acceleration and entropy terms, f_1^X can be defined as follows:

$$f_1^X = [(u/\cos^2\alpha - a)f_2^X - uf_3^Y/\cos^2\alpha - a(f_1^Y + f_2^Y)]/(u/\cos^2\alpha + a) \tag{2.45}$$

In particular, if the left boundary is vertical,

$$f_1^X = \frac{u-a}{u+a}\, f_2^X \tag{2.46}$$

Resetting – The above procedure is analytically correct; it may produce disturbing effects in a numerical computation, if continued for thousands of steps. In fact, (2.45) provides a correct evaluation of the time derivatives of a and u at the left boundary (that is, an evaluation assuming that a_0 is constant in time and space). The *updating* of a and u at the left boundary according to (2.45), however, is affected by truncation errors and, in a long run, such errors may accumulate resulting in minor, but not insignificant, variations in the effective value of a_0. The inconvenience can be overcome by resetting a_0 to its exact value after each updating, and recomputing a, u and v accordingly. From (2.42) and (2.43), u turns out to be defined by the equation:

$$\mathcal{A}u^2 + 2\mathcal{B}u + \mathcal{C} = 0 \tag{2.47}$$

with

$$\mathcal{A} = \delta(\delta + 1/\cos^2\alpha), \quad \mathcal{B} = \delta^2 R_2^X, \quad \mathcal{C} = \delta^2(R_2^X)^2 - a_0^2$$

[the + sign must be used in front of the square root in solving (2.47)]; then, a follows from (2.43) and

$$v = -u \tan\alpha \tag{2.48}$$

. When the flow is supersonic or resetting is applied, the derivatives computed on the left boundary are not used to integrate the equations of motion on the left boundary, but the values of f_i^X's and f_i^Y's must be evaluated at the first level of a two-level scheme, because they are needed at the second level, as seen later on.

B) *Upper boundary* – We distinguish two geometrical possibilities. If the upper computational is horizontal, it will be considered as a rigid, *inviscid* wall, along which $f_3^Y = f_4^Y = 0$ and f_2^Y is given by

$$f_2^Y = f_1^Y \tag{2.49}$$

to satisfy the inviscid boundary condition, $v = 0$.

If the upper computational boundary is curved, with the concavity upwards, it is considered as an entry boundary for the flow. Therefore, if the flow is supersonic, all values are prescribed and kept invariant in time at the end of the accelerating phase. If the flow is subsonic, a routine similar to the one explained for the subsonic left boundary will be applied, interchanging the roles of u and v. Specifically,

$$f_2^Y = [(v + a\sin^2\alpha)f_1^Y + vf_3^X + a(f_1^X + f_2^X)\sin^2\alpha]/(v - a\sin^2\alpha) \tag{2.50}$$

and, for resetting, (2.47) can be used, with

$$\mathcal{A} = \delta(\delta\sin^2\alpha + 1), \quad \mathcal{B} = -\delta^2 R_1^Y \sin^2\alpha, \quad \mathcal{C} = [\delta^2(R_1^Y)^2 - a_0^2]\sin^2\alpha$$

C) *Lower boundary* – The lower boundary is composed of two parts: a free-stream region to the left of the leading edge of the wall, and the wall itself. The former is always a horizontal segment and it is consider fully inviscid. Therefore, it acts as an inviscid rigid wall, with v vanishing identically, and the only numerical condition is

$$f_1^Y = f_2^Y \tag{2.51}$$

At any point on the wall, $u = v = 0$. If the wall is isothermal, its temperature, T_w is prescribed. If it is adiabatic, T_w equals the computed value of T at the grid point just above. The pressure is also considered equal on any wall point and the point above it, in any case. Therefore, if the wall is isothermal, a follows from its definition (1.5) and S from the constance of p, using (2.4) and (1.5):

$$S_\eta = a_\eta/(\delta a) \tag{2.52}$$

If the wall is adiabatic, $S_\eta = a_\eta = 0$. Thus, all quantities at the wall are easily known and no computation of f_i^X's or viscous terms is necessary. In a two-level technique, f_2^Y, f_3^Y and f_4^Y at the wall are needed at the second level for the grid row immediately above the wall to provide a second-order correction, since v is normally positive. They may, however, be taken equal to their values on the row above the wall.

D) *Right boundary* – On the right boundary, a zero-th order extrapolation for all variables can be used. In a supersonic zone, the extrapolation is obviously harmless, because no signal propagates upstream. Inside the subsonic zone of a boundary layer, the approximation is acceptable since the flow is essentially parabolic. The only dubious case occurs with a practically inviscid subsonic boundary, where f_2^X at points immediately next to the boundary is affected by the values of a and u at the boundary itself. In the computation, however, the error can be minimized by letting $f_2^X = 0$ both at the boundary and on the grid column immediately before it.

3. OUTLINE OF THE COMPUTATIONAL CODE, IN THE ABSENCE OF SHOCKS

In the absence of shock, the computational code is structured as follows:

The main program calls a subroutine `ELDATA` that reads the input data, necessary to define the geometry of the rigid walls and to generate a proper grid, the value of M_∞, and the Reynolds and Prandtl numbers according to their definitions (1.14) and (1.15). Other data are parameters to define the coordinate stretchings between ξ and X and between η and Y.

Then it calls a subroutine `GRID` that generates the grid and provides double arrays with the values of G, ϕ_1, ϕ_2, $\cos\alpha$ and $\sin\alpha$, and single arrays with the values of X_ξ and Y_η. The values of G are obtained numerically as

$$G = |\Delta\zeta/\Delta z| \tag{3.1}$$

using centered differences. Similarly, α is obtained as

$$\alpha = \arctan(\Im\Delta z/\Re\Delta z) \tag{3.2}$$

using centered differences. Finally, ϕ_1 and ϕ_2 are obtained as

$$\phi_1 = G_\xi/G, \quad \phi_2 = -G_\eta/G \tag{3.3}$$

and the derivatives of G are approximated numerically, using centered differences, evaluated on the (X,Y)-plane and interpreted in terms of ξ and η via X_ξ and Y_η.

The field is initialized as a gas at rest, with non-dimensionalized values of the thermodynamical parameters as mentioned above for the conditions at infinity ($T = 1$, $a = \sqrt{\gamma}$, $S = 0$). Then the sequence of computational steps is started.

In it, $q_{\infty t}$ is computed; a subroutine VISC is called to compute V_s and V_m; next, (2.25), (2.26), (2.27) and (2.28) are integrated at the predictor level in a subroutine PRED. The entire procedure is repeated, but PRED is replaced by a subroutine CORR for the integration at the corrector level. This completes one computational step. Boundary conditions are enforced by calling a subroutine BOUND both from PRED and from CORR.

We outline now the four subroutines used in each step.

PRED contains a loop to define $\lambda_i^X, \lambda_i^Y (i = 1,2,3)$, and $R_i^X, R_i^Y (i = 1,2)$. The time step size is evaluated according to the CFL rule; if necessary, a stability coefficient, less than 1, is used to reduce it. A second loop computes $f_i^X, f_i^Y (i = 1,2,3,4,5)$ using averaged values of λ-s over two adjacent points and two-point differences, both from the side from which the pertinent signal comes. In defining the f-s, each one is divided by 8. A factor of 2 is due to averaging the λ-s; another factor of 2 is the one in front of the brackets in (2.26), (2.27) and (2.28); the third factor of 2 makes the updating take place over a half time step only.

Successively, the boundary conditions are enforced. Then, a third loop defines S_t, a_t, u_t and v_t, in that order, using the formulas:

$$\begin{aligned} S_t &= .5V_s + f_5^X + f_5^Y \\ a_t &= \delta(f_1^X + f_2^X + f_1^Y + f_2^Y + aS_t + \delta aV_s) \\ u_t &= f_1^X - f_2^X + f_3^Y + .5(q_{\infty t}\cos\alpha + V_u) \\ v_t &= f_1^Y - f_2^Y + f_3^X - .5(q_{\infty t}\sin\alpha - V_v) \end{aligned} \tag{3.4}$$

and updates S, a, u and v. Resetting of all boundary values is made. In a final loop, all the f-s are stored in "old" arrays, f_{1o}^X, etc. The f_i^X of the left boundary are also stored in the "old" arrays for the extra row to the left; similarly, the f_i^Y of the upper boundary are stored in the "old" arrays for the extra row above it. Such values will be used in CORR to correct values on the boundaries and are acceptable in all supersonic cases on the left boundary. On the upper boundary, if curved, λ_1^Y is generally positive; therefore, f_1^Y must be corrected in BOUND. Similarly, f_2^X must be corrected in BOUND on the left boundary, if the flow is subsonic

CORR follows the general pattern of PRED. The time step, however, is no longer evaluated. All the f-s are now divided by 4 only, and their "old" counterparts, computed in PRED, are subtracted from them. Note that such "old" values must be taken at the neighboring point, from the side from which the signal proceeds No storage of f-s into "old" arrays is now necessary.

VISC begins by defining T as a function of a at all grid points, and by linearly extrapolating u, v, a, T and S at the grid column outside the computational field, to the right of the right boundary This is made to enable centered differences to be taken at boundary points, using the same scheme as at interior points For the same reason, u, a and T are defined symmetrically, and v is defined antisymmetrically at the lower boundary in front of the wall. Along the wall, u and v are defined antisymmetrically, whereas a, T and S are extrapolated linearly.

Then, at all points except at the upper boundary, centered differences are used to approximate u_ξ, v_ξ, T_ξ, u_η, v_η and T_η; μ is computed as a function of T according to (1.24) and (2.31), (2.32), (2.35) and (2.38) are evaluated. Using a function $k(T)$ as defined in (1.25), $Q_1 = GkT_\xi$ and $Q_2 = GkT_\eta$ are computed. Again, τ_{11}, τ_{12} and Q_1 are linearly extrapolated outside the right boundary, τ_{22} and Q_2 are made symmetric, and τ_{12} is made antisymmetric across the wall.

Finally, at all points except on the lower and upper boundaries, centered differences are used to evaluate $(\tau_{11})_\xi$, $(\tau_{12})_\xi$, $(Q_1)_\xi$, $(\tau_{12})_\eta$, $(\tau_{22})_\eta$ and $(Q_2)_\eta$. After this, all the elements needed to build V_s and $\mathbf{V}_m$ are available. The viscous terms are made equal to zero at all grid points to the left of the grid line passing through the leading edge of the wall.

BOUND redefines the f's at the boundaries when necessary. As mentioned before, no correction is necessary on the left boundary, if the flow is supersonic. If it is subsonic, f_1^X is redefined according to (2.45). On the upper boundary, f_2^Y is always redefined according to (2.50), if the boundary is curved. Otherwise, (2.49) is used. At the right boundary, f_2^X is set equal to zero if the flow is subsonic. At the lower boundary, to the left of the wall, (2.51) is used. On the wall, $f_1^Y = f_2^Y - \mathbf{V}_m \cdot \mathbf{j}/2$; this term is only used to produce a value of f_1^Y to be used in CORR at the grid row above the wall.

4. COMPUTATION OF SHOCKS

Shocks are considered as sharp discontinuities. Each shock is, thus, a line separating a low-pressure region from a high-pressure region. At each point on a shock, a normal to the shock can be characterized by a unit vector, $\mathbf{N}$. All values on the low-pressure side of a shock will be denoted by an index, A, all values on the high-pressure side by an index, B.

In what follows, it is convenient to consider three set of unit vectors at each shock point:
1) $\mathbf{I}$ and $\mathbf{J}$, parallel to the Cartesian axes x and y,
2) $\mathbf{i}$ and $\mathbf{j}$, parallel to the grid lines ξ and η,
3) $\mathbf{N}$ and $\mathbf{T}$, normal and parallel to the shock.
According to (2.14),

$$\mathbf{I}\cdot\mathbf{i} = \cos\ \alpha,\quad \mathbf{I}\cdot\mathbf{j} = -\sin\ \alpha,\quad \mathbf{J}\cdot\mathbf{i} = \sin\ \alpha,\quad \mathbf{J}\cdot\mathbf{j} = \cos\ \alpha \tag{4.1}$$

Let ψ be the angle between $\mathbf{N}$ and $\mathbf{I}$, θ the angle between $\mathbf{N}$ and $\mathbf{i}$. Then,

$$\theta = \psi - \alpha \tag{4.2}$$

The two components of $\mathbf{N}$ along $\mathbf{i}$ and $\mathbf{j}$ will be called N_1 and N_2, respectively:

$$N_1 = \mathbf{N}\cdot\mathbf{i} = \cos\ \theta,\quad N_2 = \mathbf{N}\cdot\mathbf{j} = \sin\ \theta \tag{4.3}$$

The velocity vector, $\mathbf{q}$, can be decomposed along the three frames defined above:

$$\mathbf{q} = u_c\mathbf{I} + v_c\mathbf{J} = u\mathbf{i} + v\mathbf{j} = \tilde{u}\mathbf{N} + \tilde{v}\mathbf{T} \tag{4.4}$$

Consequently,

$$\tilde{u} = uN_1 + vN_2,\quad \tilde{v} = -uN_2 + vN_1 \tag{4.5}$$

and, conversely,

$$u = \tilde{u}N_1 - \tilde{v}N_2, \quad v = \tilde{u}N_2 + \tilde{v}N_1 \tag{4.6}$$

The motion of a shock is defined by the velocity (W) of each of its points along the local normal. The expression:

$$M = \frac{\tilde{u}_A - W}{a_A} \tag{4.7}$$

is the normal, relative Mach number of the shock at each point; here it will be called the shock Mach number, for brevity. The Rankine-Hugoniot conditions can be written in the form:

$$\begin{aligned}
a_B &= a_A \frac{[(\gamma M^2 - \delta)(1 + \delta M^2)]^{1/2}}{(1+\delta)M} \\
\tilde{u}_B &= \tilde{u}_A + a_A \frac{1 - M^2}{(1+\delta)M} \\
\tilde{v}_B &= \tilde{v}_A \\
S_B &= S_A + \frac{1}{2\delta\gamma} \ln \frac{\gamma M^2 - \delta}{1+\delta} - \gamma \ln \frac{(1+\delta)M^2}{1+\delta M^2}
\end{aligned} \tag{4.8}$$

We define

$$\Sigma = (a_B + \delta|\tilde{u}_B - \tilde{u}_A|)/a_A \tag{4.9}$$

From these equations we obtain

$$\Sigma = \frac{1}{(1+\delta)M}[\sqrt{(\gamma M^2 - \delta)(1 + \delta M^2)} + \delta(M^2 - 1)] \tag{4.10}$$

which yields the basic relationship between Σ and M; consequently, if Σ is known, M can be computed from (4.10). For computational purposes, it is convenient to use a relationship between an increment in M and the corresponding increment in Σ:

$$\Delta\Sigma = \frac{\Delta M}{1+\delta}[2\delta + \frac{2\gamma\delta M^2 + \gamma - \delta^2}{\sqrt{(\gamma M^2 - \delta)(1 + \delta M^2)}} - (1+\delta)\frac{\Sigma}{M}] \tag{4.11}$$

In the framework of the computational technique exposed in Section 2, we observe:

a) once we have the basic variables, a, S, $\tilde{u}$ and $\tilde{v}$ on the low-pressure side of the shock, and the shock Mach number, the shock problem is solved; in fact, (4.8) provide the values on the high-pressure side and (4.7) provides the velocity of the shock;

b) the values of a, S, u and v on the low-pressure side of the shock depend only on upwind values and are accurately defined using the λ-scheme with upwind differences only;

c) knowledge of the direction of the normal to the shock is imperative to get $\tilde{u}$ and $\tilde{v}$ from u and v, and to get u and v on the high-pressure side of the shock from $\tilde{u}$ and $\tilde{v}$; moreover, it is necessary to move the shock point correctly, once W has been determined;

d) the shock Mach number is obtained from Σ, using (4.10) or (4.11), but Σ must be obtained from the low- and high-pressure side of the shock.

By writing (4.9) in the form:

$$\Sigma = \delta[a_B/\delta - \tilde{u}_B + \tilde{u}_A]/a_A \tag{4.12}$$

and comparing with the last of (2.23), we see that (4.12) can be written as:

$$\Sigma = \delta(\tilde{R}_{2B} + \tilde{u}_A)/a_A \tag{4.13}$$

where $\tilde{R}_{2B}$ is to be computed using downstream information, since the flow in the high-pressure side is subsonic, relatively to the shock. The evaluation of Σ is thus feasible without using differences across the shock.

Clearly, the most difficult step in fitting a shock consists of determining the direction of the normal. When a shock point is detected, we consider the jump in **q** between the two points bracketing it, and define the normal by its unit vector as

$$\mathbf{N} = \frac{\Delta bfq}{|\Delta bfq|} \tag{4.14}$$

The same procedure has not been proven generally safe to evaluate the normal at a shock point that is part of an existing link of shock points. In this case, we prefer to use a more cumbersome procedure. For a given shock point, located on the m-th horizontal row to the right of the n-th vertical column, we explore some intervals to the left and the right of it on both the $m-1$ and the $m+1$ row. If no points are found, the shock point is considered isolated and it is dropped. If some point is found, the slope of the shock is computed as the slope of the line joining the geometric center of the points on the lower row and the geometric center of the points on the upper row, if the flow is subsonic downstream of the shock. If the shock is supersonic downstream, the line is drawn between the shock point itself and the geometric center of the points on the row that contains the "upstream" shock point.

The shock is computed only at the end of each computational step, after all the values at grid nodes have been updated by predictor and corrector. The values at points A and B on the shock itself cannot be updated; it is indeed in the nature of a numerical scheme on a discreet grid to let all points that are not nodes remain undefined. A simple way to circumvent the difficulty is to assume that such values are the same as the values at the nearest node on either side (0-th order extrapolation). The approximation is generally sufficient for unsteady flows; it may prevent convergence of residuals to machine-zero in steady flows asymptotically reached, if a shock point happens to lie too close to a grid node. The shock point may periodically move from one grid cell to the next, and back again; at each transition, the extrapolated values may be sufficiently different to create a local disturbance that propagates downstream in successive steps. The difficulty is eliminated by using a more sophisticated extrapolation technique to define point A and point B. For example, for point A, located to the right of grid point n, we can define:

$$\begin{aligned} \epsilon &= |z_s - z_n|/|x_{n+1} - x_n| \\ \epsilon_1 &= |z_s - z_n|/|x_n - x_{n-1}| \\ \epsilon_2 &= |z_s - z_{n-1}|/|x_{n-1} - x_{n-2}| \end{aligned} \tag{4.15}$$

and then

$$\begin{aligned} a_1 &= a_n + \epsilon_1(a_n - a_{n-1}) \\ a_2 &= a_{n-1} + \epsilon_2(a_{n-1} - a_{n-2}) \\ a_A &= \epsilon a_1 + (1 - \epsilon)a_2 \end{aligned} \tag{4.16}$$

and similarly for S, u and v. It is easy to see that the values at A do not change if the shock point moves from the right boundary of cell $(n, n+1)$, with ϵ close to 1, to the left boundary of cell $(n+1, n+2)$, with ϵ close to 0.

The values at A must now be considered as the final updated values for the computational step on hand. The values at B still have to be updated after the shock itself has been updated. Only the combined quantity, $\tilde{R}_2$ is acceptable so far. We proceed, thus, to get

$$\Delta\tilde{u} = (u_B - u_A)N_1 + (v_B - v_A)N_2 \tag{4.17}$$

and

$$\Sigma = (a_B + \delta|\Delta\tilde{u}|)/a_A \tag{4.18}$$

Now, (4.11) may be applied to get the increment of M through the time-step from the increment of Σ; having M, (4.8) are applied to get new values at B; u_B and v_B will be decoded from $\tilde{u}_B$ and $\tilde{v}_B$ using (4.6). Final values at the node next to B ($n+1$, say) are obtained from formulas such as

$$a_{n+1} = a_B + \epsilon_3(a_{n+2} - a_B) \tag{4.19}$$

with

$$\epsilon_3 = |z_{n+1} - z_s|/|z_{n+2} - z_s| \tag{4.20}$$

The shock velocity is computed from (4.7). The displacement of the shock point along a ξ-line is obtained as

$$\Delta\xi = W\frac{G}{N_1}\Delta t \tag{4.21}$$

In general, the formation of a shock by coalescence of characteristics is detected as follows:

On each ξ-line, the first difference of a and the second difference of λ_2 (briefly, λ) are approximated at each point n by

$$\Delta a = a_{n+1} - a_{n-1} + 2(a_{n+2} - an - 2) \tag{4.22}$$

and

$$\Delta_2\lambda = -\lambda_{n+1} - \lambda_{n-2} - \lambda_{n-1} - \lambda_n + 2(\lambda_{n+2} + \lambda_{n-3}) \tag{4.23}$$

respectively. These expressions have been obtained by replacing the data with their linear mean-square approximations, to eliminate the influence of minor numerical irregularities. If the flow goes from supersonic to subsonic (that is, if $\lambda_n > 0$ and $\lambda_{n+1} < 0$) and $-\lambda_{n+1}/\lambda_n > 1.1$, the point on the X-line where the interpolated λ vanishes is marked. Other points marked are those where $\Delta_2\lambda_n < 0$, $\Delta_2\lambda_{n+1} > 0$ and $\Delta a > 0$. If any of such points is found along the X-line, a tentative value of Σ is defined using q_{n+1} and q_n instead of $\tilde{u}_B$ and $\tilde{u}_A$, and a_{n+1}, a_n instead of a_B, a_A The point is assumed as a new shock point, provided that $\Sigma > 1.03$ and no other shock points exist on the cells between $n-4$ and n.

If a shock is generated by a wedge or a sharp corner in a rigid wall, a third-degree equation is solved iteratively to get the direction of the normal to the shock at its point of origin With σ being the square of the sine of the flow deflection, τ the square of the cosine of the angle between the normal and the upstream direction of the flow, $M_A = u_A/v_A$, $d_1 = 1/M_A^2$, $b = -1-2d_1$, $c_1 = 2d_1 + d_1^2$, $c_2 = (1+\delta)^2 + 2\delta d_1$, $d_2 = -d_1^2$, the relation between the known σ and the unknown τ is

$$\sigma = \frac{d + \tau[c_1 + \tau(b + \tau)]}{d + \tau(-c_2 + \gamma\tau)} \tag{4.24}$$

The equation has a meaningful solution only if the shock is attached, in which case all the jump conditions at the shock point are evaluated analytically and kept frozen during the entire calculation. The absolute value of the angle between the normal and the upstream direction of the flow is arctan $\sqrt{(1-\tau)/\tau}$ and its sign is opposite to the sign of the deflection. Since the shock remains anchored to the corner, its velocity vanishes identically. Therefore, the normal Mach number is M_A, multiplied by the cosine of the angle between the impinging velocity and the normal. All the downstream values follow as shown above. Otherwise, the shock produced by the wedge is detached and it is evaluated using the same procedure as for any other shock point, with the only constraint of the normal being parallel to the wall.

III. SAMPLE COMPUTATIONAL RESULTS

5. FLOW OVER A WEDGE

The code is going to be tested for cases of increasing complexity, both for inviscid and viscous flows. In principle, a single code may be used in all cases. Substantial changes, however, must be made to the initial and boundary conditions, and it is convenient to use separate codes when the changes require more than a few IF statements.

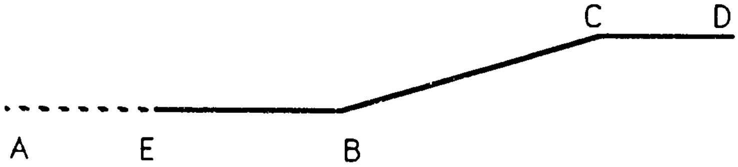


Fig. 5.1 – First wall geometry

For all cases, an orthogonal grid is used. We have used two simple definitions of the wall geometry:

First geometry (Fig. 5.1) — The lower boundary contains a rigid surface composed of three parts: a straight, horizontal segment between a point E and a point B, a straight segment between point B and point C, and an arc (almost straight and horizontal), between point C and point D. To the left of point E, the boundary contains a straight segment, AE, that is considered permeable for viscous calculations and impermeable for inviscid calculations. In the latter case, point E has no relevance and it is not considered in the code. The orthogonal grid is obtained as follows. Let ψ be the wedge angle, that is the angle between BC and the horizontal x-axis. Inputs for the geometry are the abscissae of points B, C and D (z_A being assumed equal to zero) and the angle ψ. Let $z = x + iy$ be the complex coordinate of the physical plane, $\beta = \exp(i\psi)$, $z = x + iy$, $z_1 = x_1 + iy_1$,

$$p_1 = \pi/(\pi - \psi), \quad p_2 = \pi/(\pi + \psi) \tag{5.1}$$

$$z_1 = [(z - x_B)/\beta]^{p_1} \tag{5.2}$$

The function defined by (5.2) maps the broken line ABC onto a straight portion of the x_1-axis, with the origin at B. Let $\zeta = \xi + i\eta$ and

$$\zeta = [\beta(z_1 - x_{1C})]^{p_2} \tag{5.3}$$

This function brings the origin to point C and flattens the angle between BC and CD, leaving the segment ABC on the ξ-axis. We define

$$\sigma_x = \xi_D - \xi_A \tag{5.4}$$

and introduce a computational variable, X, running from 0 to 1:

$$X = (\xi - \xi_A)/\sigma_x \tag{5.5}$$

Along the η-axis, we apply a stretching, introducing a stretching parameter, σ_y and a computational variable Y defined by:

$$\eta = Y_1[1 + \frac{\tanh \sigma_y(Y-1)}{\tanh \sigma_y}] \tag{5.6}$$

where Y_1 is a suitable value, to define the thickness of the physical region. Consequently,

$$X_\xi = 1/\sigma_x, \quad Y_\eta = \frac{\tanh \sigma_y}{\sigma_y Y_1} \cosh^2[\sigma_y(Y-1)] \tag{5.7}$$

A rectangle, $0 \leq X \leq 1, 0 \leq Y \leq 1$, is then mapped onto a domain that has a lower boundary as shown in Fig. 5.1, and the grid in the physical plane is orthogonal.

To have a ξ=constant grid line passing through point B, after the number of intervals along the X-axis has been chosen, the value of Δx is slightly modified, without changing σ_x; consequently, the position of D in the physical plane is slightly modified from the value initially chosen. No ξ=constant grid line is required to pass through point C.

After corresponding values of z and ζ have been determined, α, G, ϕ_1 and ϕ_2 are computed numerically, using centered differences.

The particular case of a flat plate with a leading edge at E is trivial.

Second geometry — A second wall geometry is similar to the one described above, but the broken line, BCD is replaced by a curve, obtained by a Kàrmàn-Trefftz mapping:

$$\frac{\zeta - 1}{\zeta + 1} = \left(\frac{z-1}{z+1}\right)^{p_1} \tag{5.8}$$

after having moved the y-axis to pass through D and having rescaled the plane to let B be at $z = -1$. No stretching is used on the x-direction, only a change of scale. Strong stretchings are used on the y-direction to account for the presence of a boundary layer. So far, we have used the stretching defined by (5.6), with a trial-and-error preliminary calculation to determine σ_y after prescribing the height of the first cells above the wall in the ζ-plane ($\Delta\eta$). A second way of defining the stretching is currently being tested. The object is to prescribe a number of intervals covering the boundary layer, according to an educated guess of its maximum height for a given Reynolds number, and to have such intervals all evenly spaced on the ζ-plane, using the stretching only above that strip. Previous experience suggests a sensible reduction in computational time being obtained by using such a device.

6. INVISCID AND VISCOUS FLOW OVER A FLAT PLATE

Basic tests require calculations of inviscid and viscous flows under the simplest assumptions. In any test, residuals are defined as mean square values of the time derivatives of u, v, a and S, as defined by (3.4). Since the code has two levels, the second of which corrects the derivatives defined by the first level, the mean square calculation is applied to the sum of the time derivatives obtained in predictor and corrector at each point; in this way, possible compensations of opposite contributions are accounted for. The logarithms (base 10) of the four residuals are plotted separately. Note that all values are reset according to the boundary conditions over the boundaries of the computational region, except at the bottom line in front of the leading edge, where the values, as updated in predictor and corrector, are accepted as valid. Therefore, no special care is taken to obtain meaningful derivatives along such boundaries and their values are not accounted for in evaluating the residuals.

We begin with a flat plate. Obviously, a test of inviscid flow starting from an exact solution would be almost insignificant because all derivatives would vanish identically. A test of an inviscid flow starting from rest, instead, is quite significant. On the flat plate the orthogonal grid is obviously Cartesian. We may, however, use an unstretched grid or a highly stretched grid. The first test (Run 343) was made on an unstretched grid with 60 intervals along ξ and 20 intervals along η, with the leading edge of the plate at $\xi = 1$; the length and height of the computational region are 5 and 2, respectively. The final $M_\infty = 2.5$ is reached after an acceleration phase lasting till $t_0 = 0.5$. "Machine-zero" values of the residuals are reached after about 500 steps (Fig. 6.1).

In the second test (Run 346), the stretching is strong ($\Delta\eta = 0.0066$). It takes longer to bring the residuals to machine-zero (Fig. 6.2) but a flow, uniform for all practical purposes, is reached in about 1000 steps. Use of a local Δt (Run 347) brings in a remarkable reduction in the number of steps (Fig. 6.3). Note that during the acceleration phase a global Δt is used, as it appears from the identity of residuals in Figs. 6.2 and 6.3 in the first phase (about 250 steps).

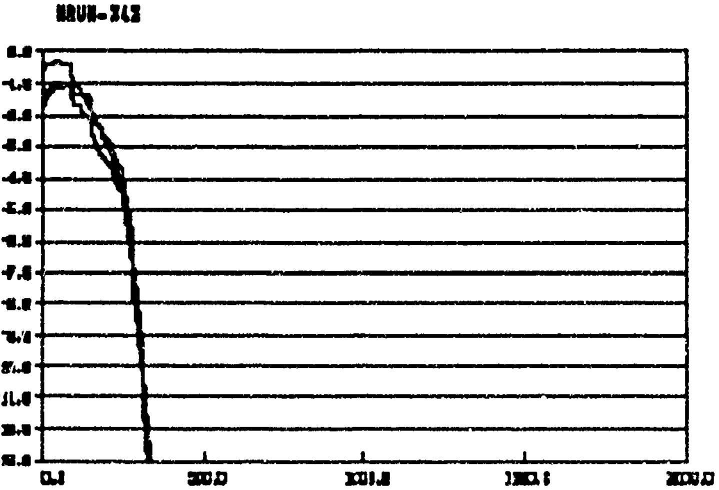


Fig. 6 1 -- Residuals for run 343

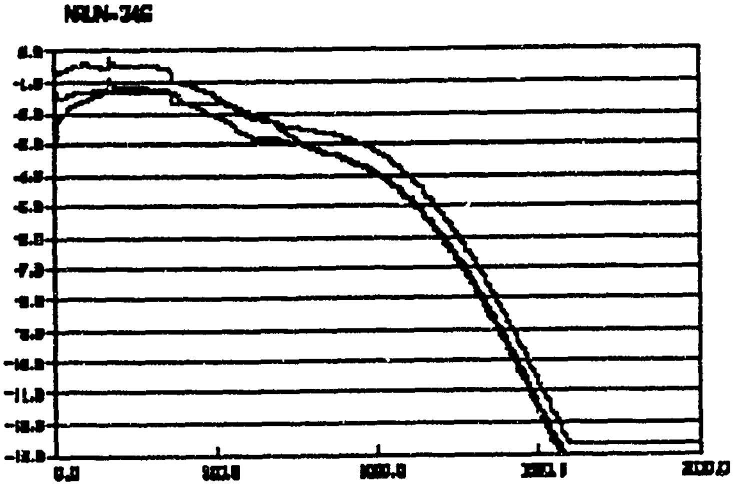


Fig. 6.2 - Residuals for run 346

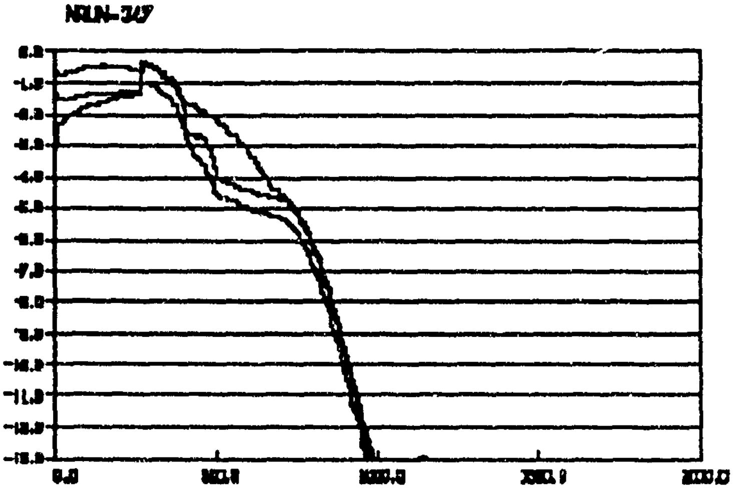


Fig. 6.3 - Residuals for run 347

For viscous flows, the most significant plots are comparisons of computed distribution of u at several cross-sections normal to the plate with the theoretical Blasius profile. Since we are dealing with compressible flows, we plot the computed values of u/V_∞ stretching the values of η not only by the factor $\mathcal{R}_e/(\xi - \xi_E)$ (where ξ_E is the value of ξ at the leading edge) but also by the factor $1/(1 + 64 * \delta M_\infty^2)$, according to Stewartson [2]. The coefficient, .64, was chosen according to Fig. 3.1 of [2]. The comparisons are generally made at three stations, called A, B and C, respectively, the first is close to the leading edge, the last is close to the right computational boundary and the second is somewhere in between.

In the following tests, the grid is the same as in inviscid cases above. The Prandtl ..umber is always equal to 1 and the plate is isothermal, with the wall temperature equal to the stagnation temperature at cruising speed. The flow is always started impulsively. As basic parameters, we consider M_∞ and $\mathcal{R}_e$.

First, we make an experiment to decide whether a global Δt can be used. For this, we choose $M_\infty = 0.2$, $\mathcal{R}_e = 1000$ and, accordingly, $\Delta\eta = 0.0066$ (run 348). We can see from Fig. 6.4 that the calculation converges, but the rate of convergence is extremely low. Some differences in the u-distributions can be seen between step 8000 and step 16000, not at station A but at stations B and C (Figs. 6. through 6.10). It seems that the braking of the flow produced by the wall takes a long time to propagate upwards, due to the fineness of the grid in the boundary layer. Therefore, the thicker the layer, the longer it takes u to reach its steady value. We repeat the run using a local Δt (Run 349). The speed of convergence increases drastically (Fig. 6.11), and the comparison with the Blasius profile is as good as with the previous run (Figs. 6.12 through 6.14). Thus, we decide to use local Δt in all these test runs. We also decide not to attempt to fit shocks, to avoid introducing new stability problems.

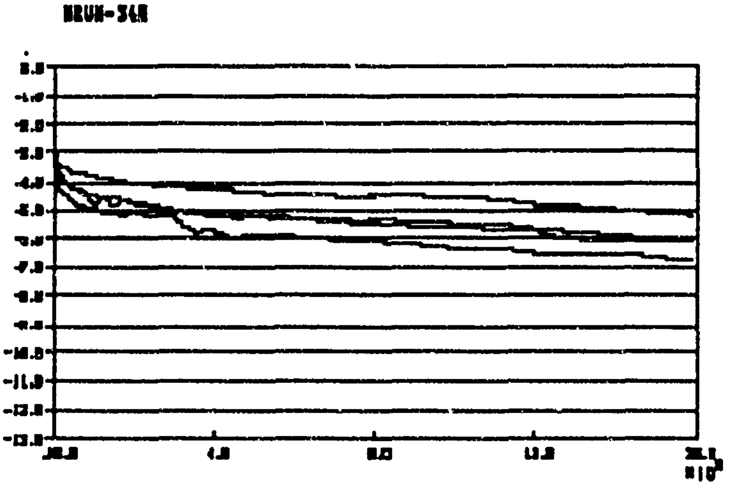


Fig. 6.4 - Residuals for run 348

Many other tests were made with different Mach numbers and higher Reynolds numbers, varying $\Delta\eta$ and Y_1 occasionally The resolution in the boundary layer seems to be sufficient, with $\Delta\eta = 0.0066$ for $\mathcal{R}_e = 1000$, .001 for $\mathcal{R}_e = 10000$ and .0005 for $\mathcal{R}_e = 100000$. A run made for the highest Reynolds number and $\Delta\eta =$ 001 showed no breaking effects propagated from the wall to the first grid line above it!

The following table shows, for all the runs made so far, an approximate value of the order of magnitude of the decrease in residuals in the first 6000 steps.

$\mathcal{R}_e$	1000	10000	100000
M_∞			
2	4	2	2
.7		4	
.9	3	7	3
1.5		11	
2	13	13	
2.5	0	13	5

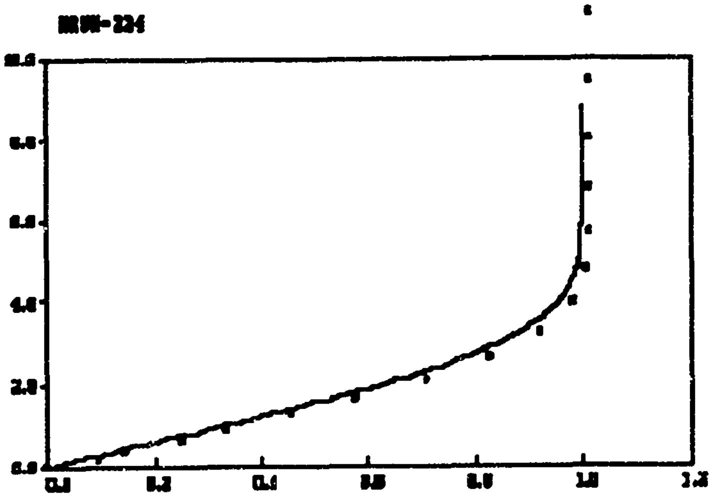


Fig. 6.5 – run 348, Blasius comparison at A, step 8000

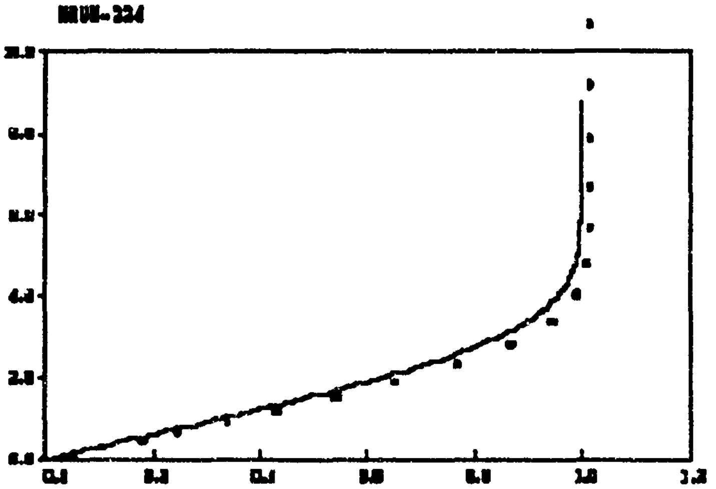


Fig. 6.6 – run 348, Blasius comparison at B, step 8000

From such results, a definite trend cannot be obtained yet, although it seems that supersonic flows converge faster than subsonic flows and runs with $\mathcal{R}_e = 10000$ converge better than others (perhaps, because the resolution in the boundary layer is more appropriate?). Runs with high Reynolds number converge very slowly and this may hamper studies of more complicated cases.

The comparison with Blasius' solution is generally very good. See, for example, Fig. 6.15 ($M_\infty = 2.5$, $\mathcal{R}_e = 10000$), Fig. 6.16 ($M_\infty = 2$, $\mathcal{R}_e = 1000$), Fig. 6.17 ($M_\infty = 2.5$, $\mathcal{R}_e = 100000$).

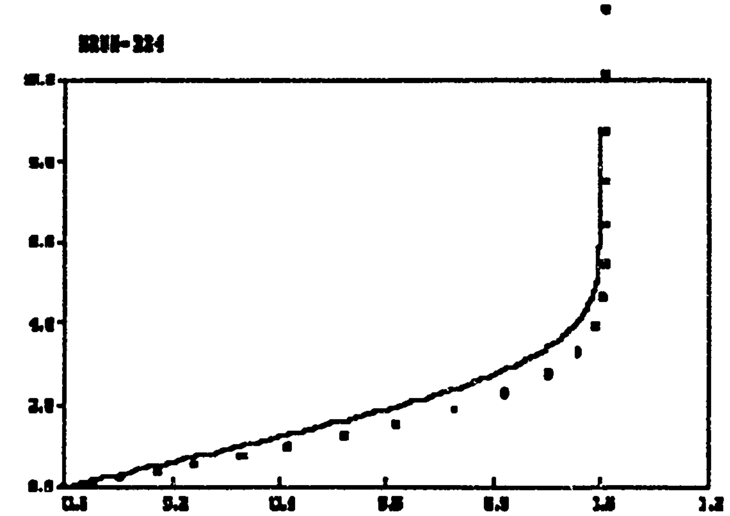


Fig. 6.7 – run 348, Blasius comparison at C, step 8000

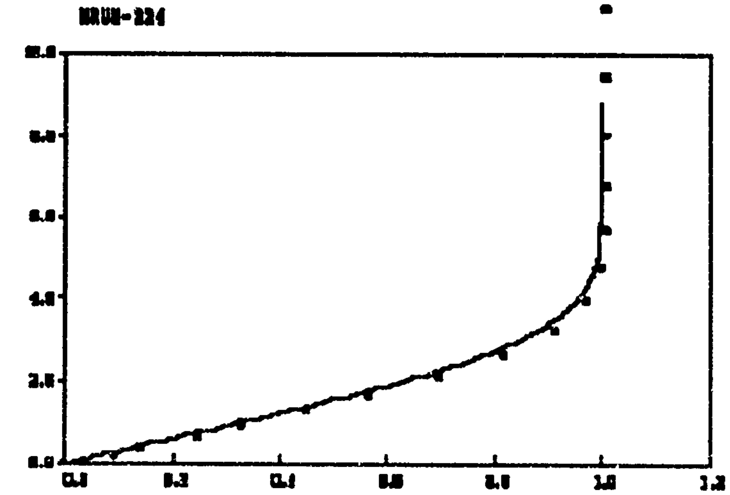


Fig. 6.8 – run 348, Blasius comparison at A, step 16000

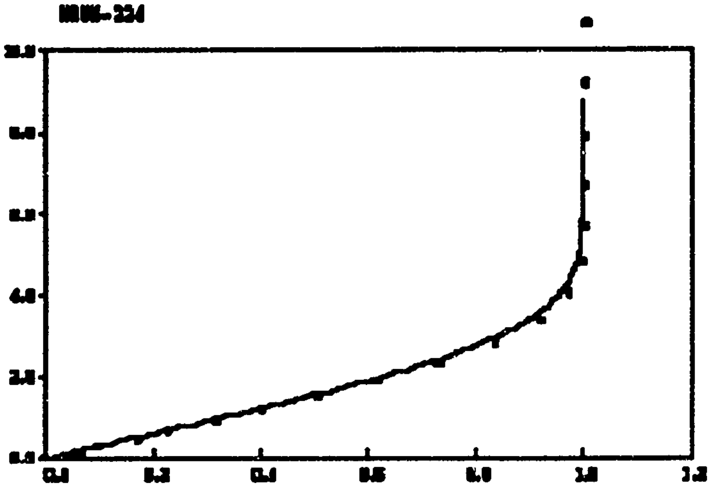


Fig. 6.9 – run 348, Blasius comparison at B, step 16000

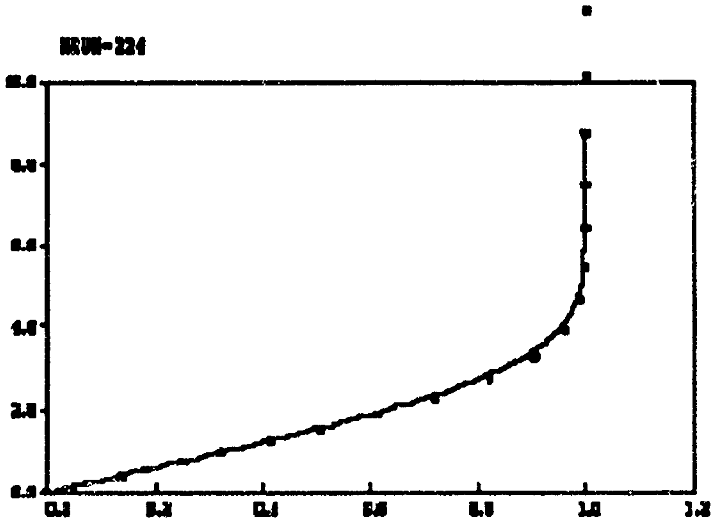


Fig. 6.10 – run 348, Blasius comparison at C, step 16000

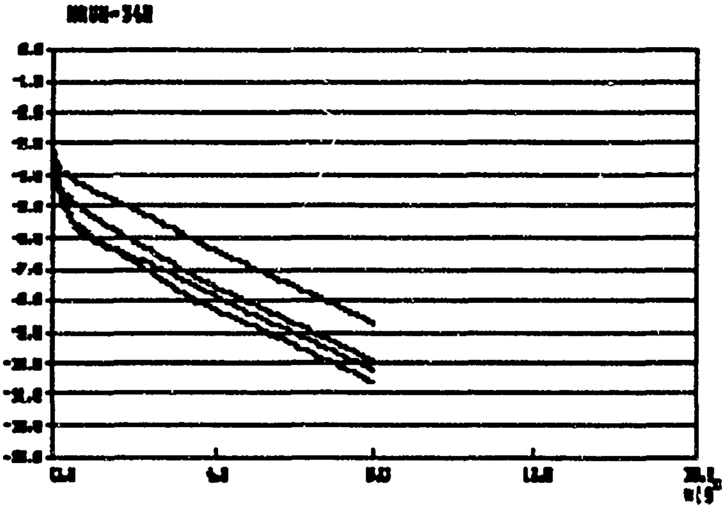


Fig. 6.11 -- Residuals for run 349

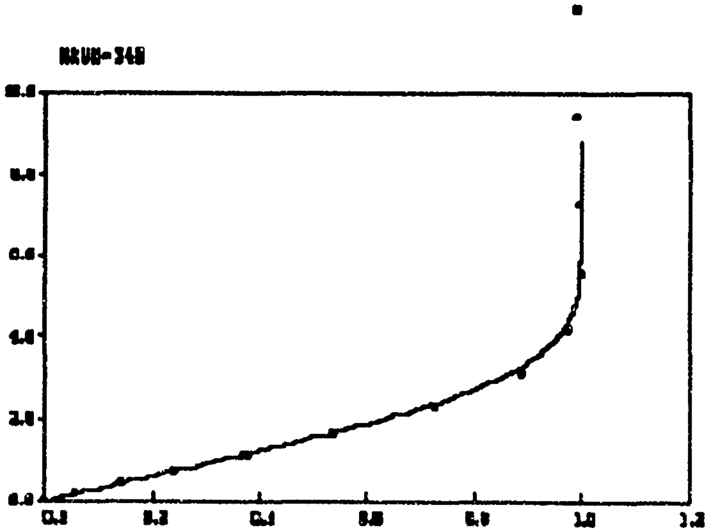


Fig 6 12 - run 349, Blasius comparison at A, step 8000

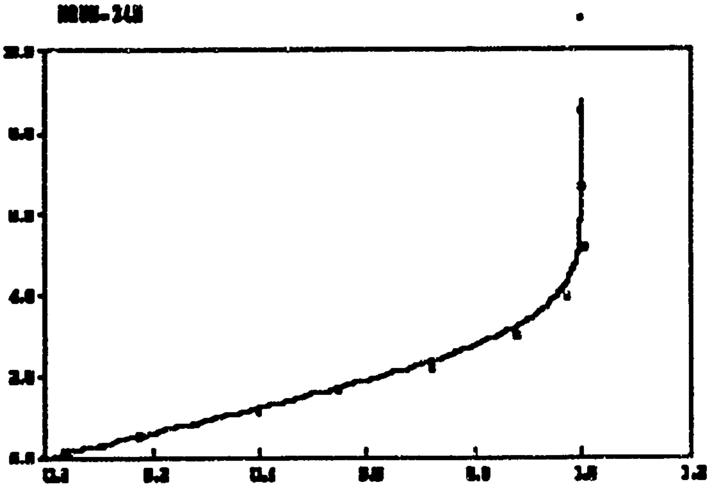


Fig. 6.13 – run 349, Blasius comparison at B, step 8000

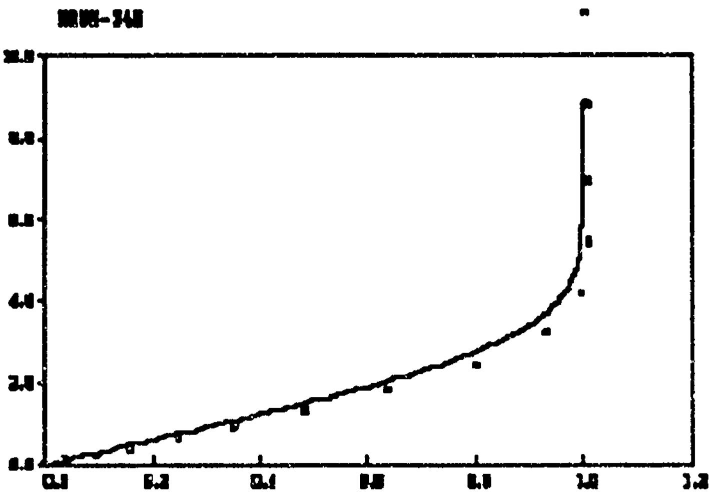


Fig. 6.14 – run 349, Blasius comparison at C, step 8000

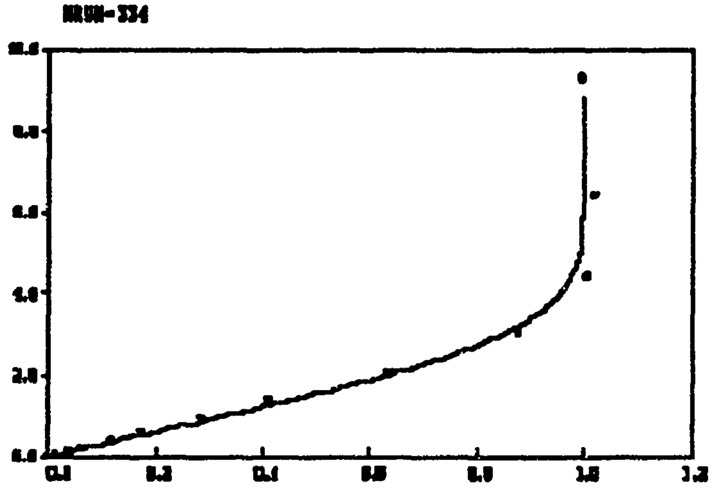


Fig. 6.15 – run 334, Blasius comparison at A, step 4000

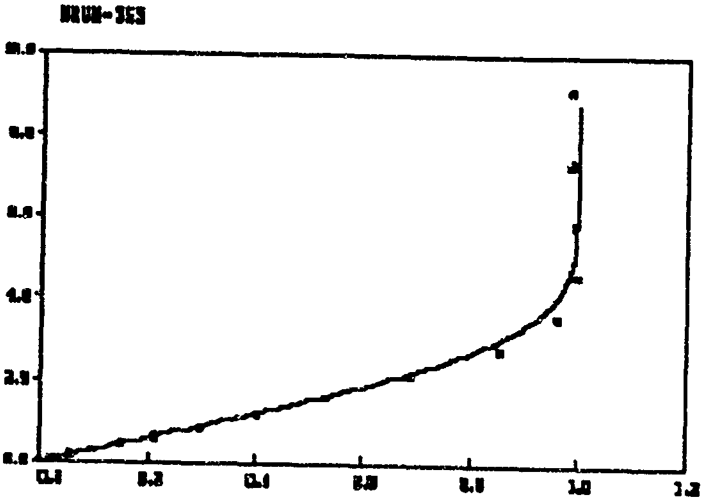


Fig. 6.16 – run 353, Blasius comparison at C, step 6000

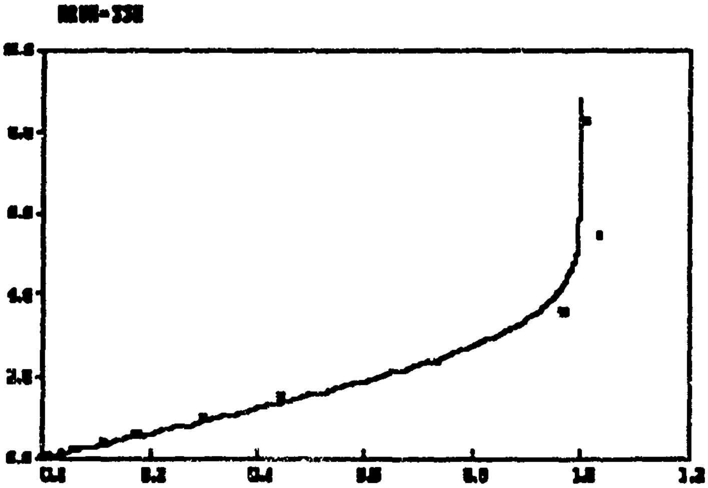


Fig. 6.17 – run 339, Blasius comparison at C, step 4000

7. INVISCID FLOW OVER A WEDGE, WITH AN EXACT SOLUTION AS INITIAL CONDITION

The first experiment in shock-fitting computes an inviscid flow over a wedge. The initial conditions are chosen to describe the exact solution. Two cases are considered, one with an unstretched grid, the other with a strongly stretched grid (as the one needed for viscous flows). The unstretched grid is poorly suited for a shock computation that uses only "x-type" shocks (that is, defined by intersections of the shocks with η=constant lines), if the free stream Mach number and the wedge angle are too high; that is because the shock tends to become parallel to η = constant lines and is defined by too few points. In this case, "y-type" shocks would be better suited.

To test the code with a stretched grid, we chose a case with $M_\infty = 6$ and $\psi = 25°$ (Runs 123 and 124). Run 123 (Fig. 7.1) cuts the computational domain before the upper corner; therefore, no expansion occurs. Run 124 (Fig. 7.2) considers the entire domain. The results at step 2000 are reasonable in both runs, but in run 124 there are signs of uncertainty at the right boundary, where the shock reaches it. This is probably due to poor handling of the shock at the right boundary. The rest of the flow field seems to be OK in both cases. The residuals (Figs. 7.3 and 7.4) reflect the inaccuracies of run 124; they decrease very slowly, with oscillations that seem to increase in amplitude. For run 123, the residuals reach machine zero in less than 2000 steps.

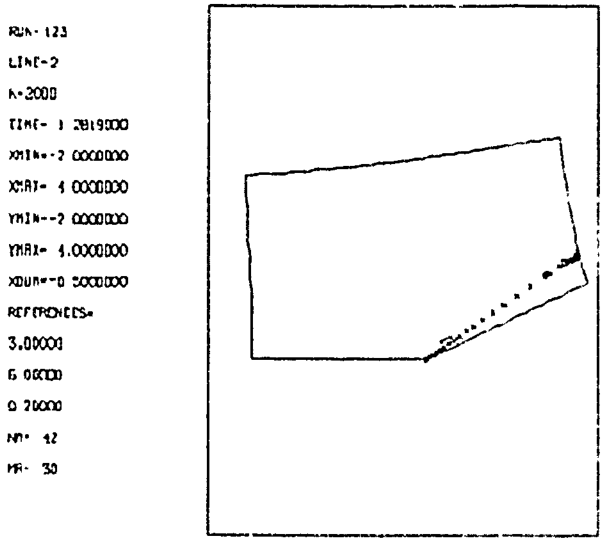


Fig. 7.1 – Isomachs at step 2000 for run 123

8. VISCOUS FLOW OVER A WEDGE, WITH AN IMPULSIVE START

To test the calculation of a viscous flow over a wedge, we focussed our attention on the second geometry, considering a flow defined by $M_\infty = 2.5$ and $\mathcal{R}_e = 10000$. We used a stretching defined by $\Delta\eta = 0.001$. Some runs were made starting impulsively ($t_0 = 0$) and running 2000 or 4000 steps using local Δt's and capturing the shocks isentropically before introducing shock-fitting. Other runs were made again starting impulsively and using shock fitting from the beginning. Interesting features were discovered during the

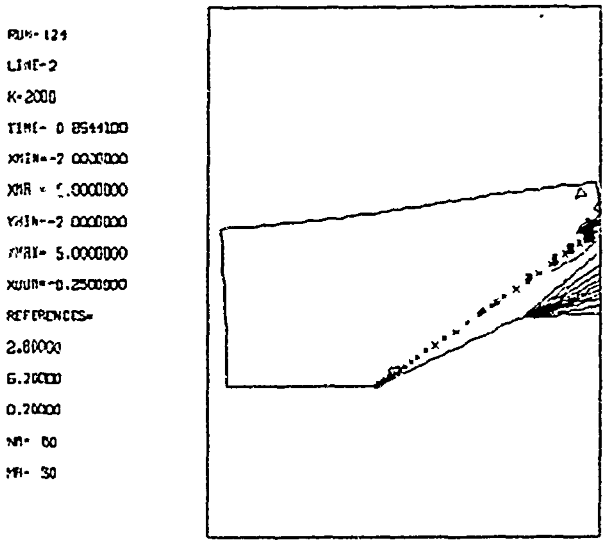
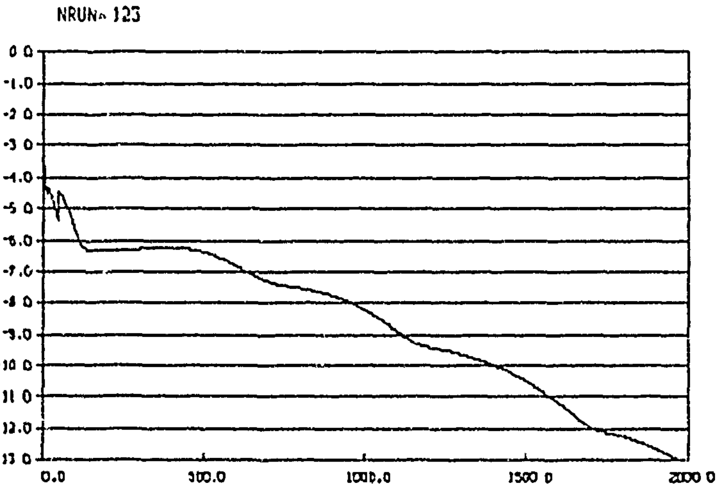


Fig. 7.2 – Isomachs at step 2000 for run 124



., 7.3 – Residuals for run 123

debugging phase. Shock-capturing runs converge to machine-zero residuals in less than 4000 steps When shock-fitting is applied, some pulsations appear in the recirculation zone and affect the rest of the flow field, interacting with the shocks that, in turn, oscillate near their origins. The upper parts of the shocks, instead, are not affected by the oscillations. So far, the cause of the oscillations has not been found yet; we believe that, most probably, it is purely numerical but a possible physical interpretation is not excluded Let us make clear that the calculations are still carried on for not more than 10000 steps and that since we must use a global Δt in the presence of shocks the physical time span actually covered is only of the order of a few millisecond. Some settling of minor oscillations in a different time scale is not at all unusual It is not

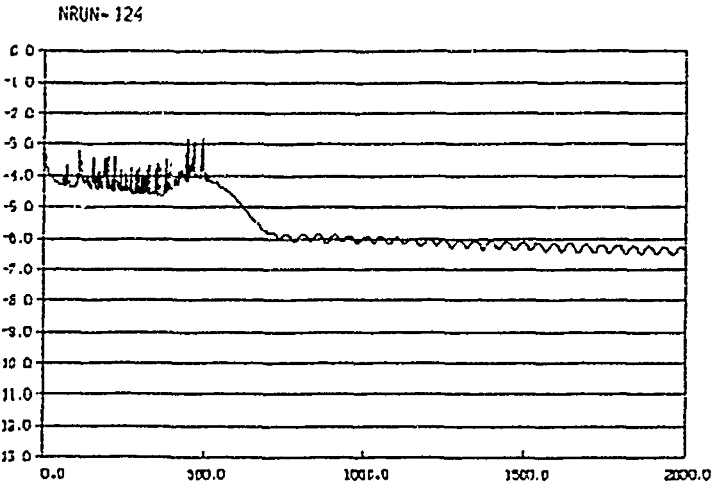


Fig. 7.4 – Residuals for run 124

considered proper, however, to try longer runs before making sure that the code is completely debugged; the logic of shock-tracking and its interactions with viscous term calculations may contain some hard-to-detect traps. Moreover, difficulties may arise if the computed region is not sufficiently high; shocks that reach the upper boundary produce boundary conditions that are hard to implement; instabilities are easily produced at the upper right corner.

Here are some results for a typical calculation. Fig. 8.1 shows a set of u=constant lines on the computational plane at step 6000. Note that the Y=constant lines are evenly spaced on the computational plane. Because of the stretching, the boundary layer occupies almost one third of the region. Only lines defined by u =0, 1, and 2 are shown in the figure. The u=0 line encloses a region of negative u, which are part of the recirculation zone. Two major shocks are denoted by crosses; the one to the left is rather weak and it is generated by the abrupt stop of the flow at the leading edge of the wall; the one at the right is stronger and it is generated by the recompression at the end of the recirculation region. The other figures show some of the features in the physical plane. Fig. 8.2 shows a few u=constant lines on the entire computed region; details of the recirculation zone are in Fig. 8.3; details of the pressure distribution in the same zone are in Fig. 8.4; note the constance of pressure on normals to the wall and the coalescence of isobars into the recompression shock.

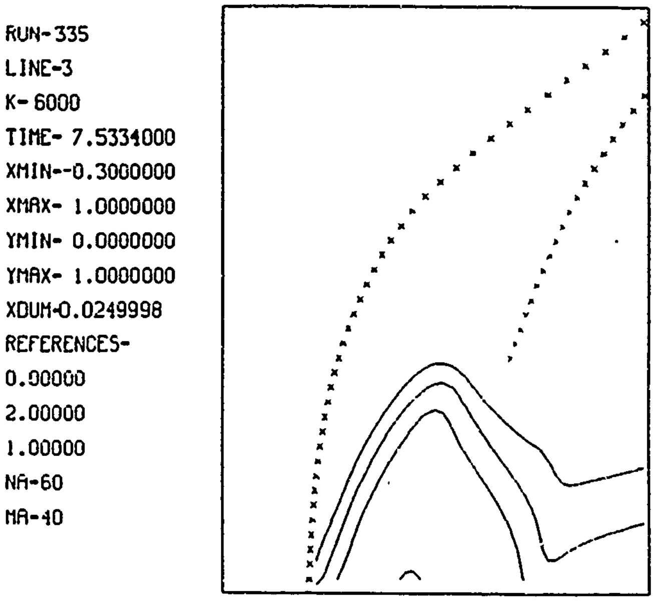


Fig. 8.1 – u=constant lines on computational plane

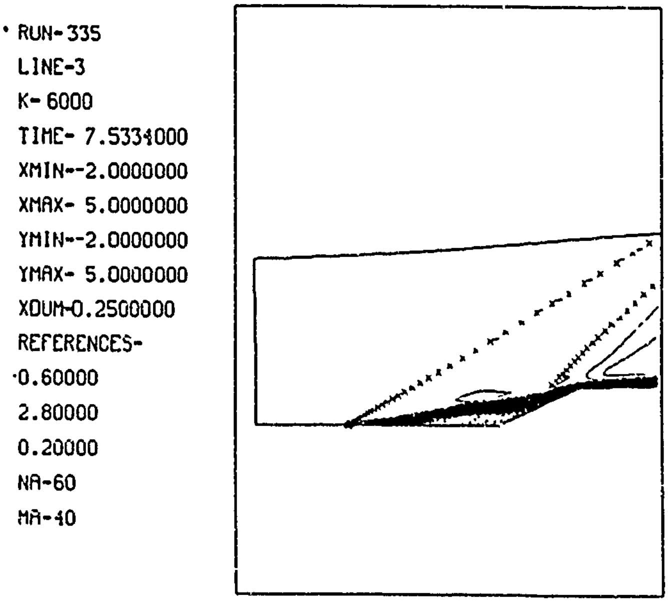


Fig. 8.2 – u=constant lines and shocks

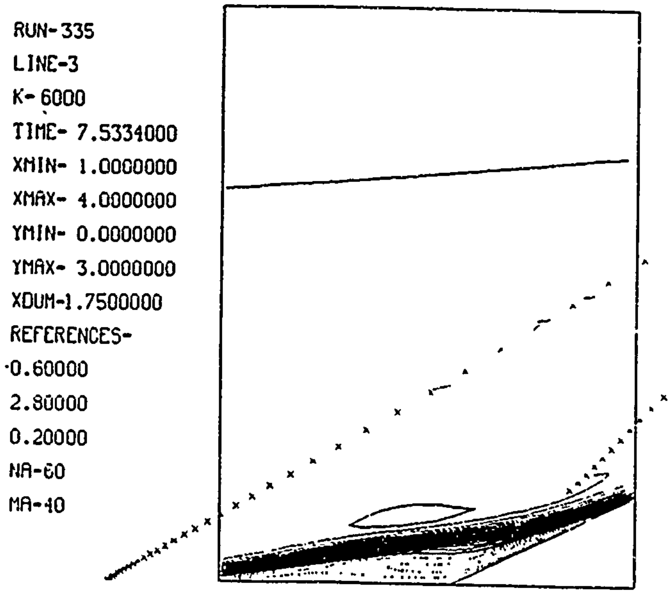


Fig. 8.3 – Enlargement of Fig. 8.2

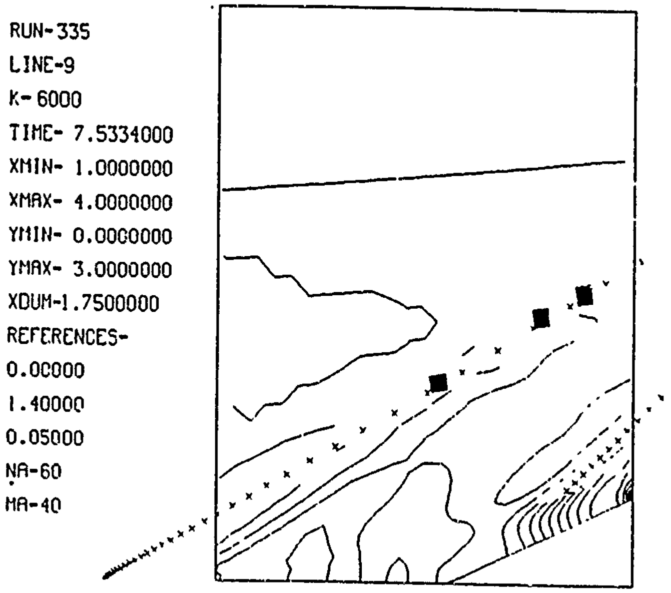


Fig. 8.4 – Isobars in the physical plane

9. SHOCK/BOUNDARY LAYER INTERACTION

The next flow considered was that produced by shock reflecting from a boundary layer. Only the steady flow will be considered. The free stream Mach number is 2 and the Reynolds number is 2.96x10^5. A down running shock is produced by a small 3° deflection. Figure 9.1 shows the isobars computed with the shock fitting code and fig. 9.2 shows the same computation with a upwind shock capturing code. The shock capturing code is state of the art in that it uses Roe's flux differencing, for this case no limiter was required so that the computation was second order everywhere. A comparison of figs. 9.1 and 9.2 indicates the amount of shock spreading in the capturing result. Figures 9.3 and 9.4 show the pressure distributions at a number of heights above the plate. While on the surface of the plate the pressures look the same off the plate the shock spreading is obvious. Figures 9.5 and 9.6 show details in the separation zone. Figure 9.5 shows the long thin separation separation bubble and fig. 9.6 shows the isobars and shocks in the same region.

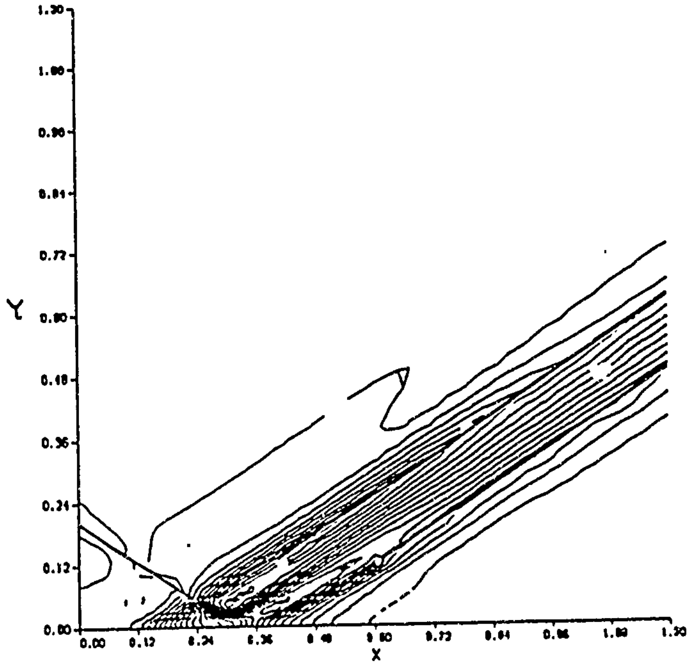


Fig. 9.1 Isobars computed with shock fitting scheme

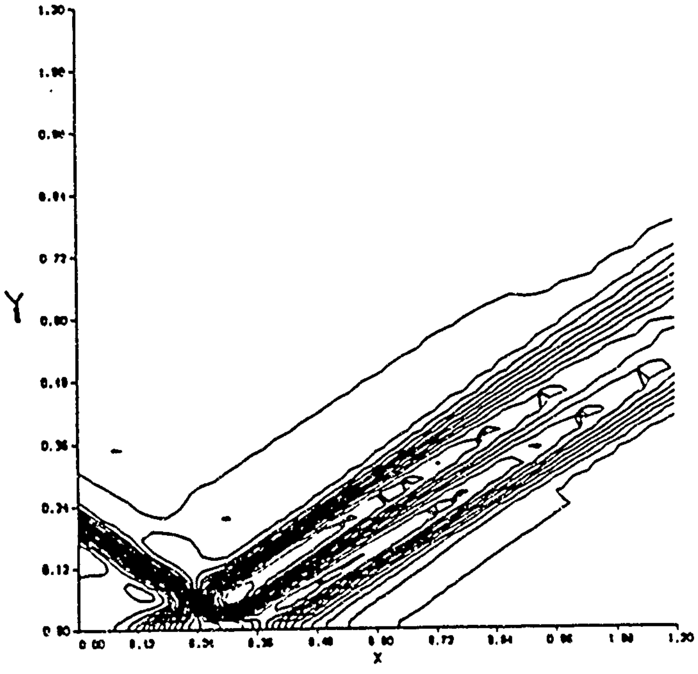


Fig. 9.2 Isobars computed with shock capturing code

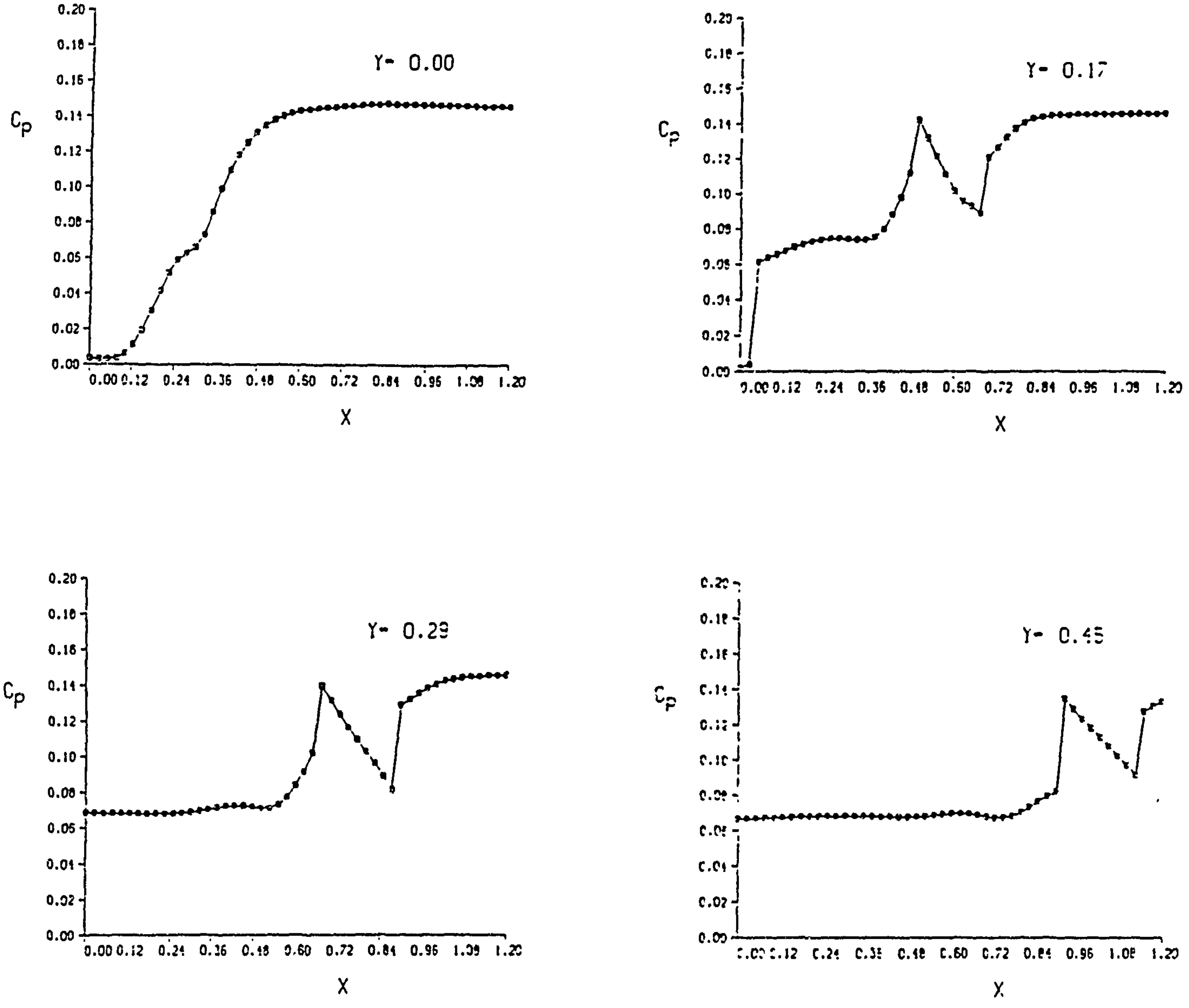


Fig. 9.3 Pressure distributions computed with shock fitting

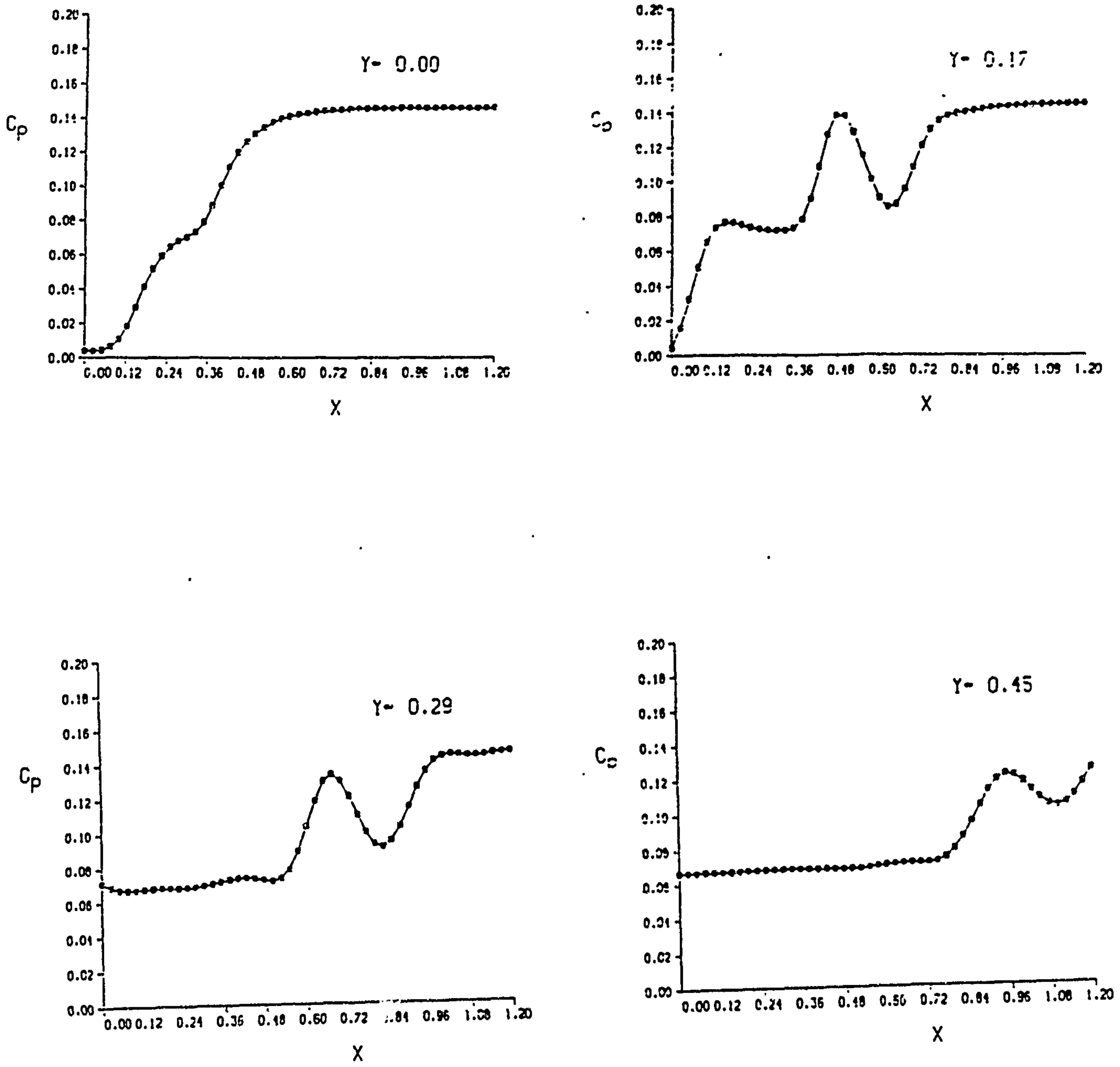


Fig. 9.4 Pressure distribution computed with shock capturing.

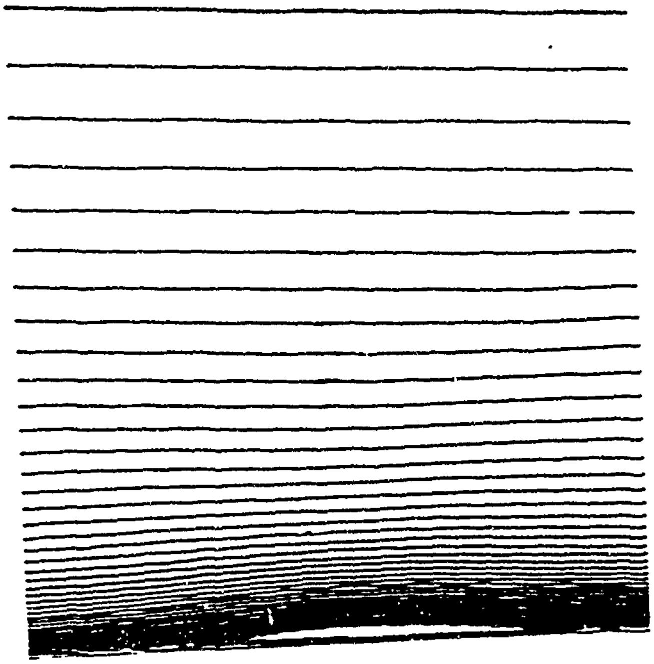


Fig. 9.5 Streamlines near separation

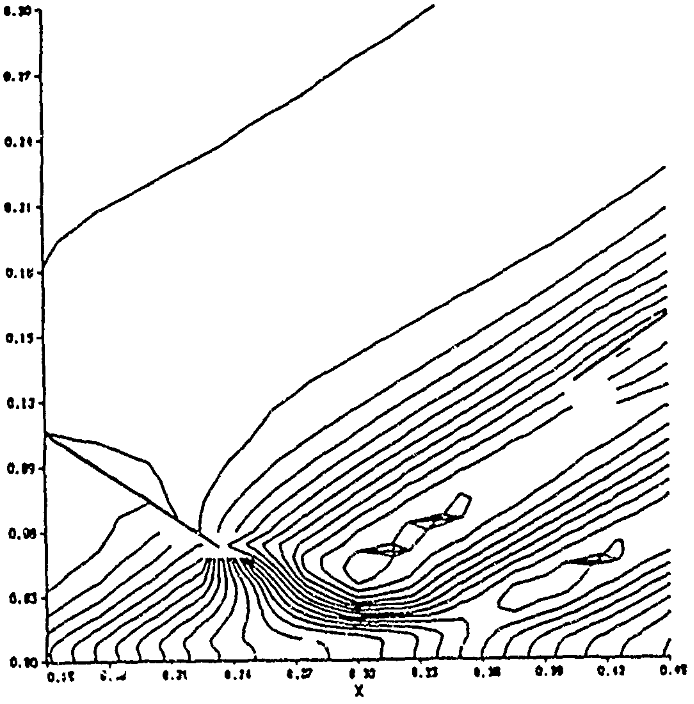


Fig. 9.6 Isobars near separation

IV. CONCLUSIONS AND FUTURE WORK

The first phase of our work has demonstrated the feasibility of performing accurate computations of shocked viscous flows. The computational scheme presented here is efficient enough for today's supercomputers, and it treats shocks in such a way as to avoid inaccuracies due to pre- or post-oscillations, spreading and local increases in numerical viscosity. The basic scheme has been demonstrated on model problems in order to prove its soundness. Any future work in this area should continue to test the scheme on flows which would tax its capabilities. In particular, the flows considered thus far were over somewhat simple geometries; future work would concentrate on more complex 2-D flows, such as the flow about a SCRAM jet inlet.

Only steady flows have been considered thus far. The code that has been developed has a time accurate capability built in so that time-dependent flows such as inlet unstart and buzz could be considered.

The impact of fitting shocks on the accuracies of the computation of shock wave/boundary layer interaction may be the most important aspect of any future work. We have begun such an investigation in the first phase of this work. An investigation of this type may uncover problems with the widely used shock capturing schemes that may have broad implications.

Only laminar flows have been considered thus far. Any investigation of shock/boundary layer interaction would be incomplete without the inclusion of turbulence effects. These investigations would start with simple models, but they would eventually examine shock/boundary layer interactions using complex turbulence models.

The real advantage of the scheme presented here may be demonstrated when three-dimensional flows are considered. While shock waves can be captured quite sharply (i.e., accurately) in one dimension, they are spread significantly in two-dimensional flows. This problem becomes even worse in three dimensions. Grid adaptive schemes can be used some what effectively to sharpen captured shocks in 2-D, but this has been not been proven in 3-D. The complexity of three-dimensional flows may require grid adaptation to obtain reliable solutions. The grid must be adapted to boundary layers, wakes, vortices, geometric irregularities (corners, etc.) and other singularities. If the shocks are fit, then that is one system of singularities that does not need grid adaptation. This is the appropriate system to fit since shocks remain discontinuous in the model we are considering. The shock is a few mean free paths thick, which is infinitesimal if a continuum flow model is assumed.

V. REFERENCES

1. Moretti, G., "A Technique for Integrating Two-Dimensional Euler Equations," J. of Computers and Fluids, Vol. 15, pp. 59-75.

2. Stewartson, K., The Theory of Laminar Boundary Layers in Compressible Fluids, Oxford, Clarendon Press, 1964.