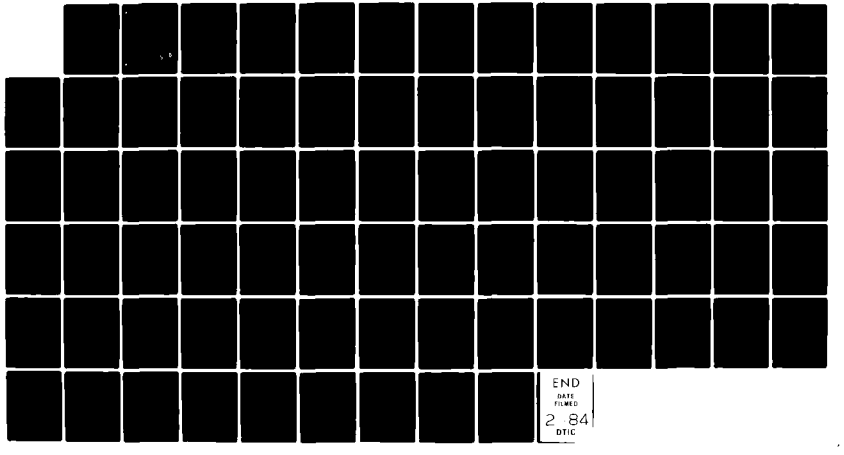
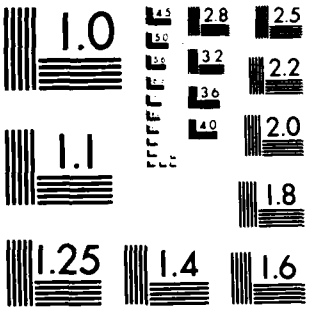
AD-A136 985 ADAPTIVE GRID GENERATION FOR NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. 1/1

UNCLASSIFIED K G BROWN DEC 83 AFIT/GAE/AA/83D-3 F/G 12/1 NL





ADAPTIVE GRID GENERATION FOR NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS

THESIS

Kevin G. Brown
First Lieutenant USAF

AFIT/GAE/AA/83D-3

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY
Wright-Patterson Air Force Base, Ohio

AFIT/GAE/AA/83D-3

ADAPTIVE GRID GENERATION FOR NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS

THESIS

Kevin G. Brown
First Lieutenant USAF

AFIT/GAE/AA/83D-3

Approved for public release; distribution unlimited

AFIT/GAE/AA/83D-3

ADAPTIVE GRID GENERATION FOR NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the Degree of
Master of Science in Aeronautical Engineering

Kevin G. Brown, B.S.
First Lieutenant, USAF

December 1983

Acknowledgments

I wish to thank the members of my committe for their support during the course of this effort, especially my advisor, Capt. Hodge. I am very fortunate to have such a distinguished committee in Capt. Hodge, Dr. Ghia of the University of Cincinnati, and Dr. Hankey of the Flight Dynamics Laboratory, all of whom are leaders in the field of Computational Aerodynamics.

Computational aerodynamics is still in its infancy. There is no doubt in my mind that, in my generation, this area will account for a major portion of the total research effort in the aerodynamics area. I hope that I can make a contribution to the field in the same way as my teachers have.

I also wish to thank Linda Stoddart whose help I could not have done without in locating the background materials for this thesis.

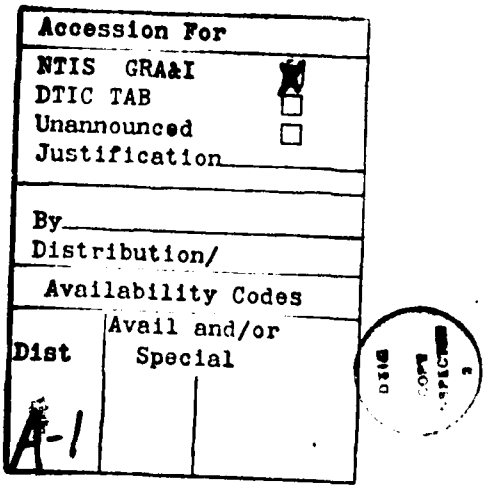


Table of Contents

List Of Figures

Abstract

A new approach for the generation of flow-adaptive grids for numerical solution of fluid dynamics problems is presented. However, the method is applicable to the numerical evaluation of any partial differential equation.

The dynamic coupling of the grid with the flow solution is accomplished through a grid-optimization technique. The optimization is based on the minimization of the finite difference truncation error in the transformed plane. The method is tested on the one-dimensional Burgers' equation which is representative of typical fluid dynamics problems. Burgers' equation is solved with an optimized SOR method using upwind differences for the convective term.

Results are presented for various Reynolds numbers and are compared to results from a similar adaptive grid method and to results for a static grid. They show the ability of the method to concentrate grid points high in gradient regions where large truncation errors occur.

List of Symbols

a_0, a_1, a_2	Least squares curve fit coefficients
f	Dependent variable
H.O.T.	Higher Order Terms
I	Total number of grid points
J	Coordinate transformation Jacobian
NP	Number of data points in curve fit
P,Q	Grid generation control functions
Re	Reynolds number
R.E.	Residual error
t, τ	Time
T.E.	Truncation error
U	Velocity
x,y	Physical space coordinates
x_τ	Grid speed
α β γ	Coordinate transformation parameters
ξ η	Computational space coordinates
ϵ_{x_τ}	Grid speed convergence criteria
θ	Relaxation factor

Subscripts and Superscripts

A	average value
M	Maximum value
i	grid point index
s	Newton iteration index
n	time iteration index

ADAPTIVE GRID GENERATION FOR NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS

I. INTRODUCTION

Fluid mechanics and heat transfer problems are characterized by complex nonlinear partial differential equations, for which analytic solutions can be obtained for only a few limited cases. Due to the rapid increase in speed and memory, and declining cost of the digital computer, an ever increasing emphasis is being placed on numerical solution of the governing differential equations by finite difference methods. In the past two decades, considerable progress has been made on the development of more efficient and accurate numerical algorithms.

A vital component of any numerical algorithm is the grid upon which the solution is obtained. The solution can be greatly simplified if a well-suited grid is chosen, just as the choice of cylindrical coordinates rather than rectangular coordinates simplifies the solution of the potential equations about a cylinder. In contrast, a poorly constructed grid can not only lead to large errors in the solution, but also can cause instability resulting in a totally incorrect solution or no solution at all. The area of numerical grid generation is relatively young receiving much attention in

the past ten years. As J. F. Thompson, a leader in the field, writes (1:1)

> This area involves the engineer's feel for the physical behavior, the mathematician's understanding of the functional behavior, and a lot of imagination, with perhaps a little help from Urania.

Two very important factors play a role in the choice of the grid. First, the coordinates should be surface oriented so that the surface boundary conditions may be implemented with no need for interpolation between grid points. This not only simplifies the boundary condition implementation but also eliminates large errors that may be produced in the interpolation process. In addition, surface oriented coordinates also permit coordinate-related approximations in the flow equations (2:35). Second, the grid must accurately resolve high gradient regions which are common in fluid dynamics problems, because it is in these regions where large numerical errors occur.

One solution of the first problem is the use of partial differential equations to generate the grid. This method was popularized by Thompson, Thames, and Mastin in 1974 (3) and is widely used today. The grid generation procedure in this thesis, presented in more detail later, is based on a modified version of the original equations presented in that paper.

The solution of the second problem is more difficult.

The method of Thompson et al. provides control for the stretching of interior regions to resolve the high gradients; however, there is, in general, no method for determining the values of those control functions due to the lack of a priori information concerning those gradients (4:28). In Ref. 4, a method was developed to compute the forcing function to minimize truncation error using boundary layer theory; the resulting mesh system is referred to as a boundary layer dependent coordinate system. Therefore, it is only valid in regions in which boundary layer theory is applicable.

In recent years, flow adaptive grids in which the grid point distribution is dynamically coupled to the developing solution have emerged as a means to tackle the second problem. Several different methods have been developed to achieve the same effect--concentrate coordinates in high gradient regions thereby decreasing truncation error and minimizing the number of grid points necessary to produce a satisfactory solution. Pierson and Kutler (5) describe a method in which the grid is defined by a minimization of the local truncation error in the least squares sense. The grid is then algebraically generated, using Chebyshev polynomials. Saltzman and Brackbill (6) define their grid based on a variational analysis resulting in a system of partial differential equations whose solution produces the grid. In the work of Dwyer et al. (7) the grid points were moved in time, based on the gradients in the

flow variable. In the paper by Ghia et al. (2), the grid adaption criterion is based on the minimization of the coefficient of the convective term in the transformed flow equations. Freeman (8) describes a method used in conjunction with the Thompson elliptic grid generation equations in which the grid control functions are determined, based on solution gradients. Anderson and Rai (9) describe another method in which the grid points move directly under an attractive/repulsive influence of one another, based on the magnitude of the local error compared to the global error. This influence can be compared to the force of a test charge in an electrostatic field or to a gravitational field which can repel as well as attract.

Anderson and Rai list the following considerations for the development of an adaptive grid (9:320):

1. The grid must evolve as part of the solution.
2. Grid points must move due to both boundary motion and changes in the interior solution.
3. The grid speed equations should be as simple as possible.
4. The grid speed equations must account for the elliptic nature of the problem.
5. The resulting grid must reduce error, provide better resolution, or otherwise improve the solution.
6. The adaptive grid scheme must be easily extended to any number of dimensions.

In addition, it is felt that the algorithm should be robust enough to handle a variety of problems with arbitrary input data.

The objective of this thesis is to develop an adaptive grid which is based on the systematic determination of the grid generation control functions used by the elliptic grid generation equations of Thompson et al. Another goal of this study is that, in the process of developing the adaptive grid, an "optimum grid" is produced which reduces truncation error, thus satisfying all conditions of item 5 above.

II. Mathematical Formulation

The adaptive grid method presented in this thesis is based on a one-dimensional analysis for ease of formulation and cost considerations. However, parts of the following analysis include the second dimension which is necessary to present a clear background for the method developed. In the following, subscripts denote partial differentiation.

Background

For complex geometries, it is convenient to transform the governing differential equations from the physical plane to a computational plane where the solution is obtained and then transformed back to the physical plane. For one-dimensional, time dependent problems, this transformation is

$$t = \tau \qquad \xi = \xi(x, t) \tag{1}$$

where x, t are the physical variables and ξ, τ are the computational variables. The derivatives in the physical plane transform as

$$f_x = f_\xi / J \tag{2a}$$

$$f_{xx} = [f_{\xi\xi} - \frac{x_{\xi\xi}}{x_\xi} f_\xi]/J^2 \tag{2b}$$

$$f_t = f_\tau - f_\xi x_\tau / J \tag{2c}$$

where if is the dependent flow variable and J is the Jacobian of the transformation given by

$$J = x_{\xi} = 1/\xi_x \tag{2d}$$

Terms involving derivatives of physical space coordinates with respect to computational space coordinates or vice versa are referred to as the metrics of the transformation and x_{τ} is referred to as the grid speed. Due to the metrics, this transformation renders the governing equations quite complex; however, there are three attractive reasons which out weigh the added complexity. First, in the computational plane, the solution can be performed on a fixed rectangular grid with uniform spacing. Second, the transformation makes it possible to concentrate the distribution of curvilinear grid lines in the physical plane in regions of high radients. Third, and most importantly, the grid lines can be made to correspond to the boundaries in the physical plane, no matter what the shape. Figure 1 shows the idea of the general coordinate transformation.

Up to now, the actual transformation has yet to be specified. One of the most popular methods for defining this transformation is the method of Thompson et al. (3). According to this method, the grid is determined as the solution of a set of elliptic differential equations

$$\xi_{xx} + \xi_{yy} = P(\xi,\eta) \tag{3a}$$

$$\eta_{xx} + \eta_{yy} = Q(\xi,\eta) \tag{3b}$$

where x, y are the physical corrdinates, ξ, η are the computational coordinates, and P, Q are functions which control the grid spacing in the interior of the region, hereafter called grid control functions.

The solution of Eqs (3) may be no easier to obtain than that of the flow equations. However, if the roles of the dependent and independent variables are interchanged so that the solution is performed in the computational plane, the boundary conditions may be specified along constant values of the computational coordinates. This results in

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = -J^2(Px_{\xi} + Qy_{\eta}) \tag{4a}$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = -J^2(Px_{\xi} + Qy_{\eta}) \tag{4b}$$

where

$$\alpha = x_{\eta}^2 + y_{\eta}^2 \tag{4c}$$

$$\beta = x_{\xi} x_{\eta} + y_{\xi} y_{\eta} \tag{4d}$$

$$\gamma = x_{\xi}^2 + y_{\xi}^2 \tag{4e}$$

$$J = y_{\xi} y_{\eta} + y_{\eta} y_{\xi} \tag{4f}$$

The Jacobian J must be non-zero for a unique transformation. For one dimension, Eqs (4) reduce to

$$x_{\xi\xi} + Px_{\xi}^{3} = 0 \tag{5}$$

Equations (4) and (5) are highly nonlinear; their solution may lead to numerical difficulties in terms of oscillations and instabilities for even moderate values of the control functions P, Q (4:21). An alernate set of grid generation equations has been proposed (11:57) which eliminates this nonlinearity. The new equations are

$$\xi_{xx} + \xi_{yy} = (\xi_{x}^{2} + \xi_{y}^{2})\, P(\xi,\eta) \tag{6a}$$

$$\eta_{xx} + \eta_{yy} = (\eta_{x}^{2} + \eta_{y}^{2})\, Q(\xi,\eta) \tag{6b}$$

which, after inversion, give

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = -(\alpha P x_{\xi} + \gamma Q x_{\eta}) \tag{7a}$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = -(\alpha P y_{\xi} + \gamma Q y_{\eta}) \tag{7b}$$

where α, β, γ are the same as in Eqs (4).

For one-dimensional problems, Eqs (7) reduce to

$$x_{\xi\xi} + Px_{\xi} = 0 \tag{8}$$

Equation (8) is the grid generation equation used in the present study. It is solved implicitly by an optimized Successive Over Relaxation (SOR) technique. One-sided upwind differences are used for the convective term based on the results of a study by Ghia, Hodge, and Hankey (4). These authors had showed that, when the control function became large, as it needs to be for large Reynolds numbers, and a central difference is used for the convective term, the solution of the grid equation becomes nonmonotonic and an oscillatory solution results. The use of an upwind difference was shown to eliminate this behavior.

Adaptive Grid Criterion

Any finite-difference representation of a derivative has truncation error associated with it. This truncation error must be small in order to obtain an accurate solution of the problem. A central-difference representation of a first derivative is

$$f_x = (f_{i+1} - f_{i-1})/ 2\Delta x + T.E. \tag{9a}$$

where the truncation error (T.E.) is given by

$$T.E. = -\frac{\Delta x^2}{3!} f_{xxx} - \frac{\Delta x^4}{5!} f_{xxxxx} + H.O.T. \tag{9b}$$

and H.O.T. stands for 'higher order terms'. Similarly, a

second-order accurate backward-difference approximation for a first derivative is

$$f_x = (3f_i - 4f_{i-1} + f_{i-2})/ 2\Delta x + \text{T.E.} \qquad (10a)$$

with

$$\text{T.E.} = -\frac{\Delta x^2}{3} f_{xxx} + \frac{\Delta x^3}{4} f_{xxxx} + \text{H.O.T.} \qquad (10b)$$

In the past, efforts to minimize trucation error have consisted of reducing the grid spacing Δx. This does indeed decrease the magnitude of the truncation error, but at the expense of an increased number of grid points necessary to cover the domain and therefore an increased computational time. For many problems, this may not be insignificant, especially for higher dimensions. An alternate method for decreasing error is to reduce the magnitude of the higher derivatives. In the physical plane, one has no control over these derivatives; however, in the computational plane, one could gain control over these terms if the transformation from the physical plane to the computational plane were based on the reduction of these derivatives.

The one-dimensional transformation of a first derivative is given by Eq (2a)

$$f_x = f_\xi / x_\xi \qquad (2a)$$

If the derivatives in the computational plane are expressed

as standard central differences with $\Delta\xi$ taken as unity, then

$$f_x = (f_{i+1} - f_{i-1}) / 2x_\xi + T.E. \tag{11a}$$

with

$$T.E. = -f_{\xi\xi\xi} / 6\, x_\xi + H.O.T. \tag{11b}$$

Thompson (1:4-6) gives a lengthy argument that T.E. must be expressed in the physical plane giving

$$T.E. = -\frac{1}{6}\frac{x_{\xi\xi\xi}}{x_\xi} f_x - \frac{1}{2} x_{\xi\xi} f_{xx} - \frac{1}{6} x_\xi^2 f_{xxx} + H.O.T. \tag{11c}$$

Equation (11c) shows the T.E. to be very dependent on the grid spacing. In short, the metrics must be minimized such that their expressions in Eq (11c) are reduced in order to reduce T.E. To reduce the metric error completely, however, would be to eliminate the advantages of the transformation.

On the other hand, if one considers the T.E. of f_ξ in the computational plane (for $\Delta\xi = 1$) then,

$$T.E. = -f_{\xi\xi\xi} / 6 + H.O.T. \tag{12a}$$

for a central-difference representation of the first derivative and

$$T.E. = -f_{\xi\xi\xi} / 3 + H.O.T. \tag{12b}$$

for a second-order backward difference. In the computational plane, the T.E. depends only on the higher order derivatives. If the solution in the computational plane were a second degree polynomial

$$f(\xi) = a_2\xi^2 + a_1\xi + a_0 \tag{13}$$

then the third, fourth, and all higher order derivatives would be identically zero. For second-order-accurate finite differences, the truncation error would be eliminated, no matter what the grid spacing, and an optimum grid would be produced. The number of grid points could then be reduced without sacrificing accuracy.

In general, Eq (13) cannot be enforced over the entire domain because there will inevitably be some error in the solution process and, if Eq (13) is not satisfied exactly, large variations in the grid spacing may cause significant errors in the solution.

It may only be necessary to enforce Eq (13) locally. The leading term of the local truncation error is proportional to the third derivative. Let it be approximated by a central difference, giving

$$T.E._i \simeq f_{\xi\xi\xi_i} \simeq f_{\xi\xi_{i+1}} - f_{\xi\xi_{i-1}} \tag{14a}$$

Again, using central differences to evaluate the second derivatives gives

$$T.E._i \approx f_{i+2} - 2f_{i+1} + 2f_{i-1} - f_{i-2} \qquad (14b)$$

Therefore, the local T.E. at a point is approximately dependent only on its immediate neighbors. In this context, the entire domain can be divided into several segments, each of which satisfy Eq (13), i.e.

$$f_i(\xi) = a_{2_i} \xi^2 + a_{1_i} \xi + a_{0_i} \qquad (15)$$

At each grid point, a second order least squares curve fit is performed using the solution at the previous iteration to calculate the constants. The number of data points NP used in the calculation may be specified to be any number greater than 3 and less than or equal to the total number of grid points. For example, if NP = 5, then the constants at each grid point i will be determined by the previous solution values at i±2, i±1, i for a balanced curve fit. At the boundaries, the curve fit will necessarily be unbalanced. The grid control function is then determined as described in the next section. If NP is equal to the total number of grid points, the entire domain is used in the least squares curve fit, producing just one set of constants. Because the constants are based on the solution at the previous iteration, the grid will change as the flow solution changes, lagging it by one iteration.

Grid Control Function Evaluation

The previous section described the goal of the adaptive grid. In this section, the means of achieving that goal is defined.

Equation (2a) gives the transformation of the first derivative. Solving for the computational derivative gives

$$f_\xi = f_x x_\xi \tag{16}$$

Differentiating w.r.t. ξ gives

$$f_{\xi\xi} = f_{xx} x_\xi^2 + f_x x_{\xi\xi} \tag{17}$$

Using the grid generation equation, Eq (8), and Eq (15), an equation of the form F(P) = 0 results where

$$F(P) = f_{xx} x_\xi - P f_x - 2 a_2 / x_\xi \tag{18}$$

for which a Newton-Raphson iteration can be performed to solve for the root P.

$$P^{s+1} = P^s - F^s(P) / \left(\frac{\partial F}{\partial P}\right)^s \tag{19}$$

where s denotes the iterate level. Using Eq (18) to determine F(P) and $\frac{\partial F}{\partial P}$ gives

$$P^{s+1} = P^s - (f_{xx} x_\xi - P f_x - 2a_2/x_\xi)/(-f_x) \tag{20}$$

At this point, a relaxation factor θ is introduced so that

the grid movement be gradual (9:321-322) and the physical derivatives are replaced by their computational counterparts, Eqs (2), giving

$$P^{s+1} = P^{s} + \theta \left[\frac{f_{\xi\xi}}{f_{\xi}} - \frac{2a_2}{2a_2\xi + a_1} \right] \qquad (21)$$

where $0 < \theta < 1$ and f_{ξ} in the last term has been replaced by its counterpart from Eq (15) for consistency.

All values at s are known for the previous time step or iteration. The constants a_2 and a_1 are evaluated from the least squares curve fit to that solution. All derivatives are evaluated as second-order central differences and initially P must be provided as input to start the solution. Currently, no formal method is used to determine θ so it must be determined by experimentation.

With the new control function specified a new grid is then determined as the solution of Eq (8). For comparison, the grid control function used by Freeman (8) is given by the first two terms on the right hand side of Eq (21). It is referred to as a linear method because it results from specifying the solution in the computational plane to be linear, therefore producing $f_{\xi\xi} = 0$. It is felt that this may be over-constraining the problem because the present analysis shows that it is only necessary to satisfy the condition $f_{\xi\xi}$ = constant, (not necessarily 0). The present case is

referred to as a quadratic method.

One-Dimensional Model Problem

The viscous Burgers' equation is chosen to test the adaptive grid method because it is typical of many problems encountered in fluid mechanics. It is a nonlinear, second order equation, and its solution produces large gradients as the Reynolds number, Re, is increased. The resulting flow can be compared to a boundary layer profile of thickness proportional to (1/Re).

The nonconservative form of Burgers' equation is

$$U_t + UU_x = U_{xx} / Re \tag{22a}$$

with boundary conditions

$$U(-\infty, t) = 1.0 \quad U(0,t) = 0.0 \tag{22b}$$

and initial conditions

$$U(x,0) = 1.0 \text{ for } x < 0 \quad U(0,0) = 0.0 \tag{22c}$$

The analytical steady state solution is given by

$$U(x) = - \tanh (x\, Re/2) \tag{23}$$

Equation (22a) is transformed to the computational

plane by use of Eqs (2) resulting in

$$U_\tau + \left[\frac{U - x_\tau}{x_\xi} + \frac{x_{\xi\xi}}{Re\ x_\xi^3}\right] U_{\xi\xi} = \frac{U_{\xi\xi}}{Re\ x_\xi^2} \qquad (24)$$

An optimized Successive Over Relaxation (SOR) method is used to solve Eq (24). For high Reynolds number flows, it has been shown that central differencing of the convective term leads to an oscillatory solution (4:23). Therefore, second-order, one-sided upwind differencing is used for the convective term. The time derivatives are expressed as a first-order backward differences and the diffusion term is represented as a second-order central difference. The metrics are evaluated as central differences.

Solution Procedure

The solution algorithm may be summarized as follows: The computer program developed to solve this problem using the adaptive grid method developed here is listed in Appendix B.

1. Provide an initial guess for the control function P.
2. Solve Eq (8) for the grid point distribution.
3. Solve Eq (24) to obtain the flow solution for the first iteration.
4. Perform a least squares curve fit of the flow solution to determine the constants a_2 and a_1.
5. Generate a new grid control function distribution by solving Eq (21).
6. Repeat Steps 2 through 5 until steady state is

reached.

Steady state is assumed to have been achieved when the average difference in the solution between two successive iterations is less than a specified tolerance. This condition is given as

$$\frac{1}{I} \sum_{i=1}^{I} | U_i^n - U_i^{n-1} | < \varepsilon \qquad (25)$$

where I is the total number of grid points, n denotes the iteration level, and ε is the tolerance.

Error Analysis

For this study, an analytical steady state solution exists, thereby providing a direct means to determine the truncation error. However, in general, no such solution will be available. Another measure of the truncation error is defined as the residual error R.E. and is determined as the difference between the computed solution and the solution defined by Eq (15). The local error, both truncation and residual, is defined to be the error at a point. The global error is the value of the local errors averaged over the entire field, i.e.,

$$T.E. = \frac{1}{I} \sum_{i=1}^{I} | U_i - U_{exact} | \qquad (26a)$$

$$R.E. = \frac{1}{I} \sum_{i=1}^{I} | U_i - U_{fit} | \qquad (26b)$$

If the solution is such that the R.E. is zero, then Eq (15) would be satisfied and the optimum grid would be defined.

III. Discussion of Results

The adaptive grid method developed in the previous section was tested by solving the one-dimensional Burgers' equation given in Eq (23). The solution of Burgers' equation exhibits an increase in slope at the right boundary and a decrease in slope in the left boundary as Re increases. For large values of Re, the solution of Burgers' equation is comparable to boundary-layer flows except that the thickness of the boundary layer is proportional to (1/Re) instead of $(1/Re)^{\frac{1}{2}}$. An essential feature of any mesh is a concentration of grid points in regions of high gradients in the flow solution. For boundary-layer flows, a good rule of thumb is to have 5 to 10 grid points inside the boundary layer thickness. It is also necessary to have a smooth variation in the distribution of grid points in the transition region from large gradients to small gradients and to have enough grid points in all regions of the domain so as to give an accurate solution of the problem. These features are used as a test of the effectiveness of the adaptive grid method developed here.

Basis of Results

It was necessary to limit the value of the control function P determined by Eq (21) because early results produced very large magnitudes of P, in some cases resulting in a

double-valued transformation violating the maximum principle. For the results presented here, that limit was set at 2.0 based on a truncation error analysis of the grid generation equation, Eq (8). For a uniform distribution of P = 2.0, I = 21, and $\Delta\xi = 1$ analytical solution of Eq (8) for B.C.s x(1) = -1 and x(21) = 0 gave x(2) - -.0067. The corresponding numerical solution using a second order upwind difference for the convective term gave x(2) = -.314. Two features are evident from this analysis. First, this is an enormous spacing which is undesirable for accurate solution of the flow equations. Second, the numerical solution is very different from the analytic solution. The truncation error analysis showed that the T.E. is proportional to exp(P). The limit value of 2.0 was chosen because the solution of Eq (22) produced much smaller values over most of the domain and local regions of P = 2.0 could be handled. An additional constraint was imposed, which provided a more uniform distribution of grid points near the wall (x = 0) boundary. This constraint was derived from the transformed Burgers' equation, Eq. (24). Because an upwind difference was used for the convective term, it was desirable to keep the value of the coefficient multiplying it positive adjacent to the wall in order to use a second-order accurate difference. For this to happen, the value of P had to be less than the product $U \cdot Re \cdot X_\epsilon$. The only place where this expression had any effect was near the

wall where the grid spacing was very small.

For this study, the grid was defined to be converged when the average value of the grid speed x_τ was less than a specified minimum ε_{x_τ}. The final converged grid depended on several factors:

1. The initial guess of the control function P which provides the initial grid. A value of zero was chosen because it produces a cartesian grid of uniform spacing. In this way, the robustness of the grid adaption mechanism could be demonstrated.
2. The number of data points NP used in the least squares curve fit which determines the constants a_2, a_1 and a_0.
3. The grid covergence criteria ε_{x_τ}. This factor was parameter dependent and will be discussed later in this section.
4. The value of the relaxation factor θ in Eq (8) plays a role in the determination of ε_{x_τ}. Its effect will also be discussed later in this section.

The infinity boundary condition for the velocity U was specifield to occur at x = -1.0. For small values of the Re, U(-1) is not necessarily equal to one, therefore it was determined from the analytic steady state solution. Results are presented here for 21 grid points in the domain $-1.0 < x < 0.0$

and values of NP = 5 and NP =21.

Grid Dependence on NP

Figures 2 through 6 present the steady state solution and the converged grid for NP =21 and for Re = 1, 10, 100, 1000, 1500. Large slopes in the grid curve indicate large spacings between the grid points, likewise small slopes in the grid curve indicate small spacings between the grid points. The solution in the transformed plane takes on a parabolic shape (as expected) for the smaller values of Re. A very good grid is produced, placing 8 and 6 grid points inside the boundary layer (B.L.) for Re = 10 and 100, respectively. However, as Re becomes large, the solution looks more like its hyperbolic tangent form in the physical plane given by Eq (23). This is because the velocity U is equal to 1.0 for all but very, very small values of x, because the boundaries are fixed, and because a quadratic function is a very poor fit to the hyperbolic tangent function.

Also note the step in the grid point distribution for Re = 1000 and 1500. This step is due to the last term of Eq (21). The coefficients a_2 and a_1 were generally of opposite signs and a_1 was generally an order of magnitude larger than a_2. For small values of ξ the sign of this term was determined by the sign of a_1; however, as ξ increased in value to approximately 8 or 9, the $2a_2\xi$ term dominated. Also, for small values of ξ , this entire term dominated

the first term, $U_{\xi\xi}/U_{\xi}$, which for large Re was equal to zero due to U being constant in value. For larger values of ξ, the $U_{\xi\xi}/U_{\xi}$ term dominated. The net effect was to produce the grid control function distribution shown in Figure 7 for Re = 1500. Positive values of P cause the grid points to move to the right, X = 0 boundary, whereas negative values of P cause movement in the opposite direction. Where P changes sign, the grid points on either side are forced together, therefore, producing the step in Figures 5 and 6. Because of this step, several grid points were placed in a "pseudo gradient" leaving only a few grid points for the real gradient. Only two grid points were inside the boundary-layer thickness of (1/Re) for Re = 1000 and for Re = 1500, only 1 grid point was inside the boundary layer.

This effect is eliminated when the least squares curve fit is performed only locally, i.e., for small values of NP. Figure 8 through 12 present results for NP = 5 for the same Reynolds numbers as before. For Re = 1, the grid did not move significantly from its original position. The pseudo gradient is no longer present for the larger values of Re because it is much easier to fit a quadratic function for only a few local data points than for the entire domain. The number of grid points inside the boundary layer is 7 for Re = 100, 4 for Re = 1000, and 6 for Re = 1500. When ε_{x_τ} was reduced to .001, 7 points were placed inside the B.L. for Re=1000. Figure 13 presents a grid control function distribution for these cases (NP=5) showing a much better dis-

tribution than the NP=21.

The number of data points used in the least squares curve fit used to determine the constants a_2, a_1, and a_0 has a significant effect on the final converged grid. Figure 14 demonstrates as NP is increased the converged grid looks more and more like that for NP =21. Even if the value of the grid convergence criteria ε_{x_τ} is decreased, the final grid will exhibit this step for the larger values of NP.

Grid Dependence on ε_{x_τ}

It is evident that the value of the grid convergence criteria is a very important factor in the shap of the final grid and in the accuracy of the flow solution. As the parameters (e.g., Re, NP, etc.) are changed, it becomes difficult to predict the value of ε_{x_τ} which will produce the best grid due to the highly oscillatory nature of the grid speed variation with time. Figure 15 shows a representative time history of the average grid speed demonstrating this feature. If too small a value of ε_{x_τ} was chosen, the grid did not converge in a reasonable number of iterations. For comparison, the solution of this model problem on a static grid required between 5 to 17 iterations to converge to steady state. Steady state solution on an adaptive grid required between 15 to 30 iterations, on the average. However, if ε_{x_τ} was too small, grid convergence was not attained even after 50 iterations. If too large a value of ε_{x_τ} was chosen, the converged grid did not fully resolve the gradients in the flow solution. Figure 16 demonstrates this behavior.

Figures 17 through 19 present a steady state error analysis based on ε_{x_τ} for two different values of Re and two values of NP. The key quantity in these figures is the maximum truncation error, $T.E._M$. It might be expected that, if the value of ε_{x_τ} was smaller, the solution would be improved. Figure 18 and 19 show this to be true. However, Figure 17 shows the contrary. At this point, no definite value can be proposed for an arbitrary set of conditions. The key quantity in these figures is the maximum truncation error, $T.E._M$. Ot might be expected that if the value of ε_{x_τ} was smaller, the solution would be improved.

Grid Dependence on θ

The value of the relaxation factor θ has an effect on on the final grid point distribution, although not as much as the other parameters. As θ was increased, more of the grid movement generated by Eq (21) was allowed to be accomplished at each iteration, i.e., the grid speed increased. This may be good, especiallyfor larger Re where more grid movement is necessary to resolve the flow gradients. Along with this increased grid speed comes an increase in the amplitude of the oscillation in the grid speed convergence, discussed previously. The smaller the value of θ, the more uniform the grid movement became. Figure 20 shows a representative effect of θ on the steady state error. In general, a better solution was achieved as θ was decreased, however,

there is a drawback. In general, the smaller ϵ was, the longer convergence took.

Comparison with Other Methods

The effectiveness of the adaptive grid method developed in this thesis is now compared to the adaptive grid method of Freeman (8) and to a static grid. An attempt was made to compare the different methods on the same basis; a key feature of the comparison is the resolution of the flow gradients. A measure of this is the number of grid points in the boundary layer ($1/Re$). Therefore, for high Re, the static grid was exponentially stretched in order to place a similar number of grid points inside the boundary layer (1/Re). As many parameters as possible wer kept the same, however, in many instances, this was not possible.

Table I summarizes the comparison of the three methods for Reynolds numbers of 10, 100, 1000, and 1500. The key parameter to note is the maximum truncation error ($T.E._M$). Freeman's method is referred to as the linear method and the present method is referred to as quadratic. For NP = 5, the adaptive methods are very similar, which is demonstrated in Figure 21. The values T. E. for the adaptive grids compare very well with the T.E. static of the grid. Note, however, that the number of grid points is substantially increased for the static grid for Re = 10 and 100 in order to resolve

TABLE I

Error Comparison

θ = 0.25 I = 21 NP = 5

Re	T.E. max			T.E. ave		
	Fixed Grid	Adaptive Grid		Fixed Grid	Adaptive Grid	
		Linear	Quadratic		Linear	Quadratic
10	.0021[a]	.0093	.0111	.0007	.0055	.0042
100	.0113[b]	.0198	.0115	.0005	.0103	.0053
1000	.0169[c]	.0227	.0291	.0049	.0084	.0087
1500	.0160[c]	.0248	.0365	.0046	.0078	.0073

a I = 51

b I = 201

c I = 21 P = 0.5

the flow gradient. For large Re, the static grid was exponentially stretched to resolve the gradient. This amounts to a constant value of P in the grid generation equation, Eq (8). This stretching was possible for this model problem because the characteristics of the solution were known, however, in general, these features will not be known a priori and a stretching of this sort would not be possible. One would then have to resort to a large number of grid points in order to resolve all the features of the flow.

Another important factor concerning the viability of any new develpment is the relative computation time required to solve the problem. This study was performed on a CDC Cyber 175 computer. For the static grid the solution of Burgers' equation required an average of 2.03×10^{-3} cp seconds per grid point per iteration. Freeman's linear adaptive grid method required an average of 3.88×10^{-3} cp seconds and the present quadratic method required 4.31×10^{-3} seconds for NP = 21 and 4.67×10^{-3} cp seconds for NP = 5. The present method required between 2 and 5 cp seconds to r reach steady state for 21 grid points compared to between .2 and .5 cp seconds for 21 grid points for the static grid. However, the static grid with 201 grid points required 3.066 cp seconds.

IV. Conclusions

The results of this study demonstrate the benefits of using an adaptive grid in the solution of fluid dynamics problems. There were two primary goals of the adaptive grid method develped in this thesis: first, to provide adequate resolution of the high gradient regions in flow; and second, to produce an optimum grid such that the truncation error would be eliminated. The first objective was met. The method does a very good job of concentrating grid points in the physical plane in high gradient regions, where large truncation errors generally occur. It showed robustness in that the gradients were resolved given an initially constant spaced cartesian grid and without any a priori knowledge of the flow. However, it is not as robust as hoped, because it depends on the input of several parameters which presently, can only be determined by experimentation. In general, the following statements can be offered for the determination of these parameters based on the results of this study.

1. The number of data points used in the least squares curve fit should be decreased as Re increases. Using the entire field for the curve fit was successful only for smaller values of Re. As the number of points is decreased, the method approaches that of the linear method used by Freeman (8).

2. Large values of the relaxation factor θ result in greater grid speeds and less convergence times. Smaller values result in decreased grid speeds and larger convergence times. However, the smaller values of θ produce better steady state results due to the dampening of the oscillations that occur in the grid speed.
3. At this point, no conclusion can be drawn about the grid convergence criteria ε_{x_τ}. It appears to be very problem dependent and can only be determined by experimentation.
4. The computation time required for solution on an adaptive grid is greatly increased over that on a static grid. However, without a priori knowledge of the flow solution, a great number of grid points is required for the static grid, putting the two methods on equal ground for this one-dimensional problem.

The second goal of the thesis was achieved only partially. The method produces satisfactory results; however, it did not produce a truly optimum grid. The truncation errors are still on the same order as those produced by a static grid with equal grid spacing or with coordinate stretching.

V. Recommendations

The results of this study indicate the advantages of the use of a solution adaptive grid for the numerical solution of partial differential equations. Further investigation of the present method is required to determine the reason the truncation errors were not minimized, as expected.

One possible source of error resided in the Newton iteration technique used to determine the grid control functions. It is suggested that a fixed Newton iteration be used; the denominator should consist of only one term, not two.

A second source of error is associated with the grid spacing. Large errors in the solution tended to occur where large grid spacings occurred, especially if the spacing was three or more times the value of the smaller grid point. The present method may need to be modified to restrict the size of the grid spacing.

Appendix A

Figures

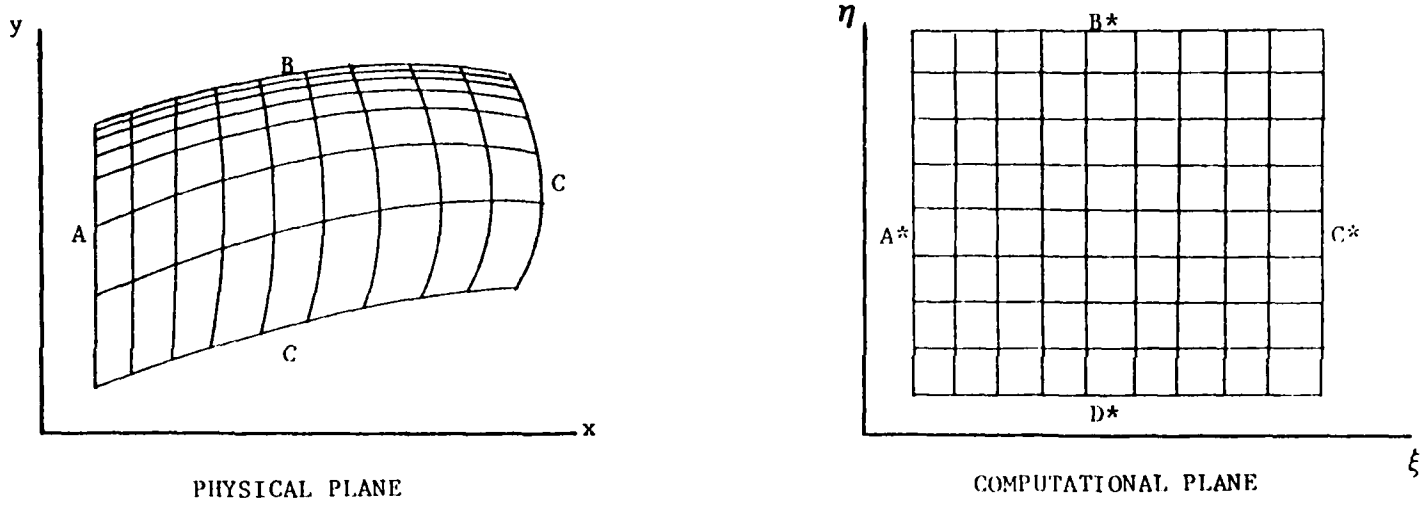


FIG 1 GENERAL COORDINATE TRANSFORMATION

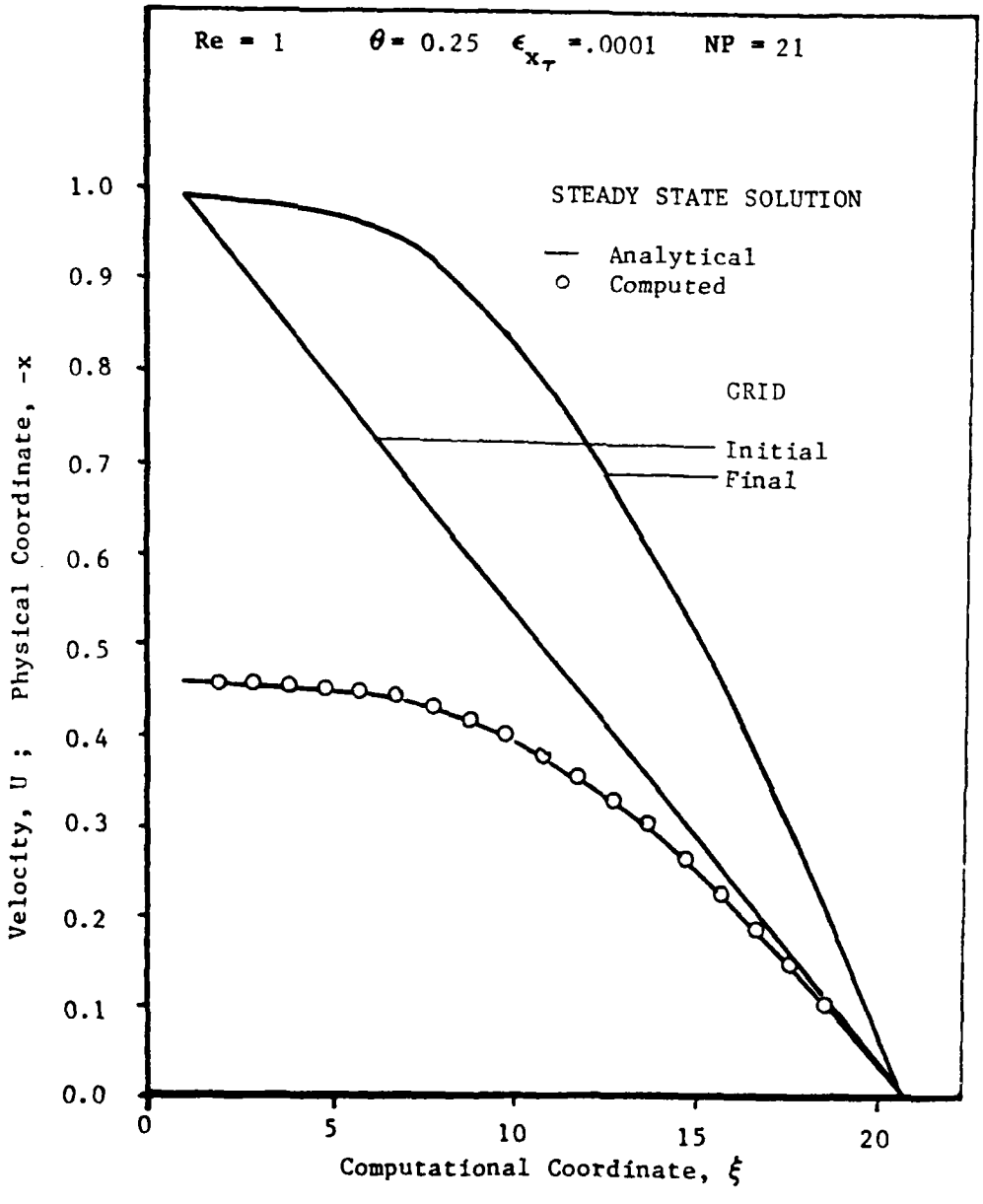


Fig 2 Velocity Profile & Grid Re = 1 , NP = 21

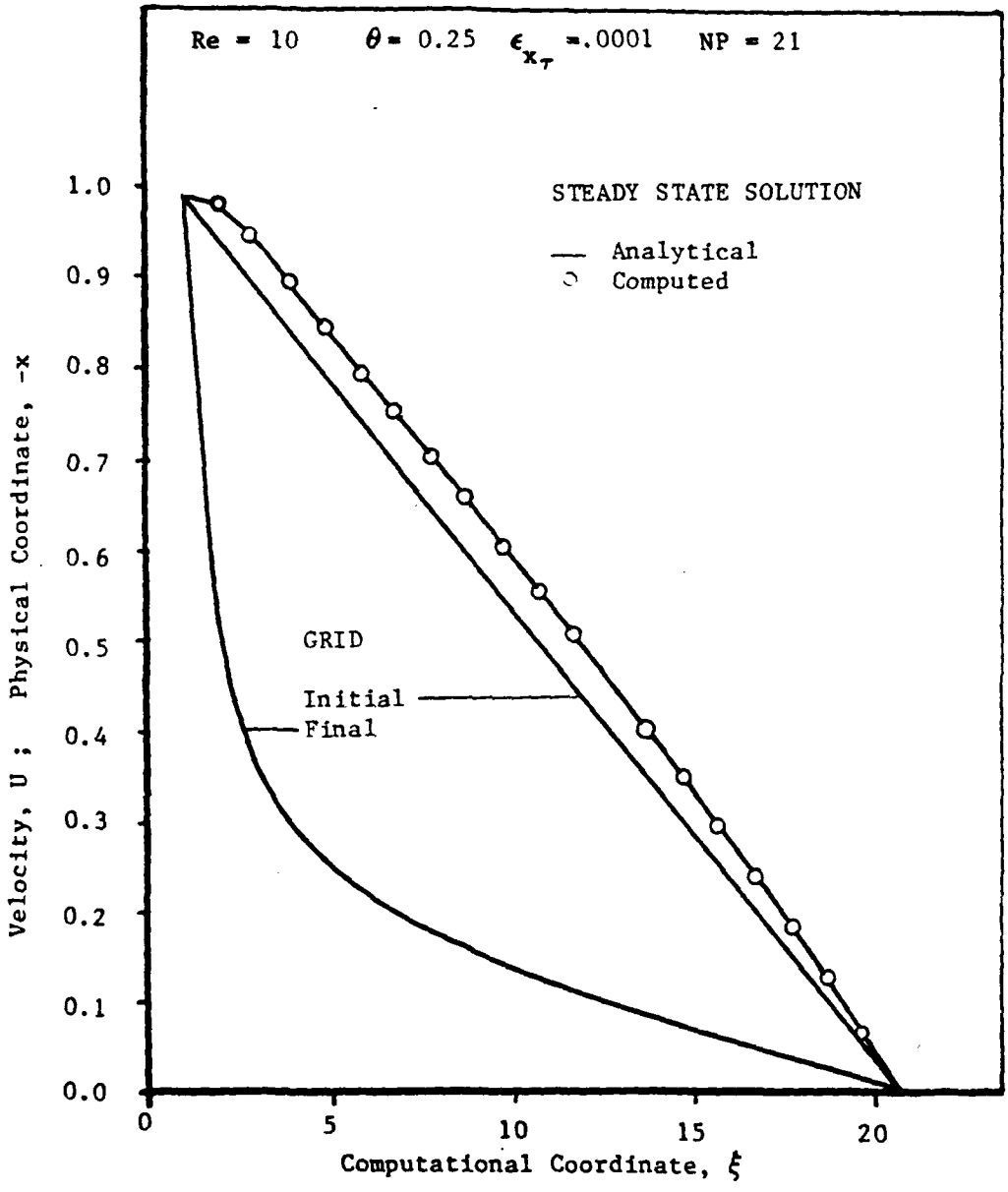


Fig 3 Velocity Profile & Grid Re = 10 , NP = 21

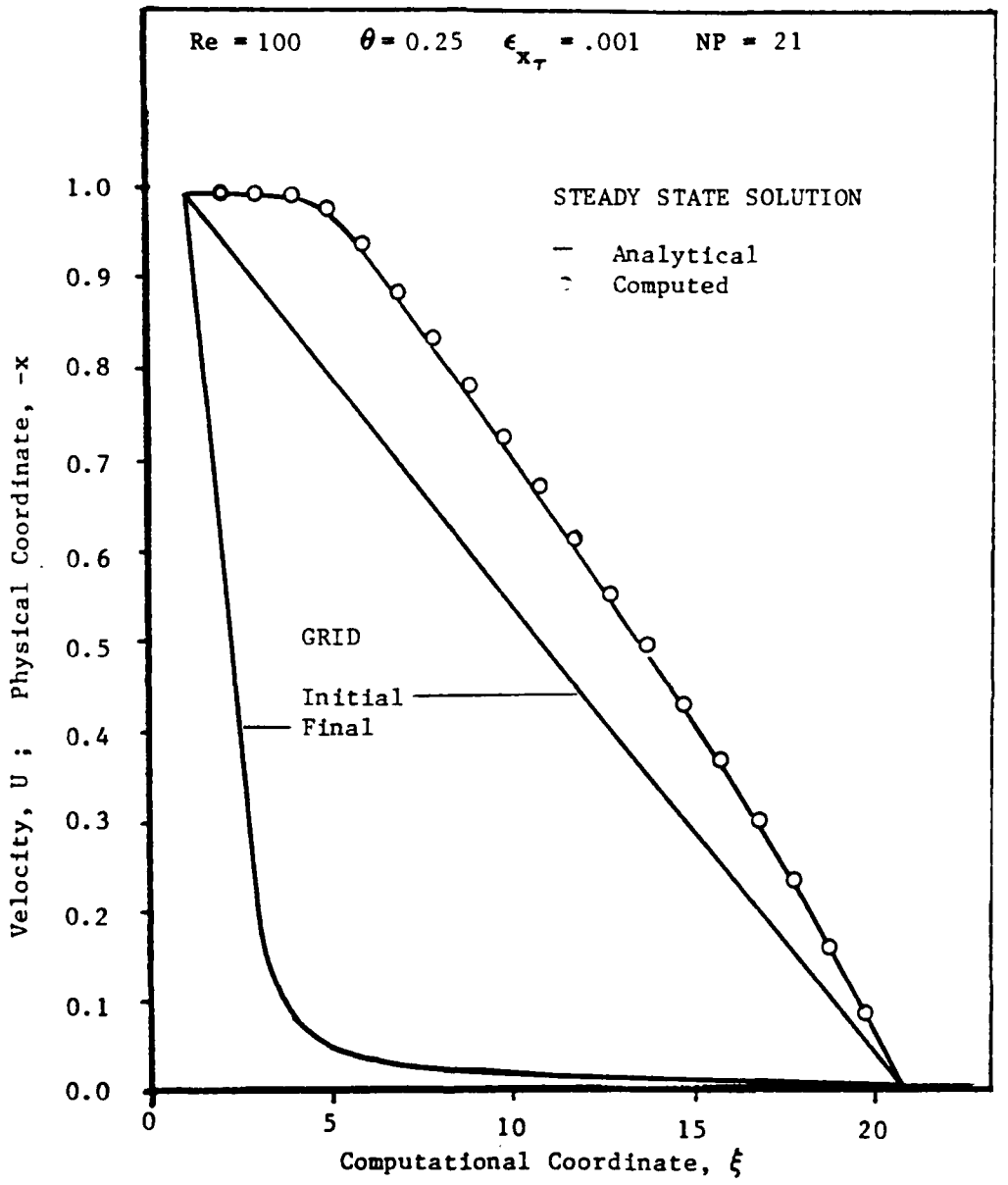


Fig 4 Velocity Profile & Grid Re = 100 , NP = 21

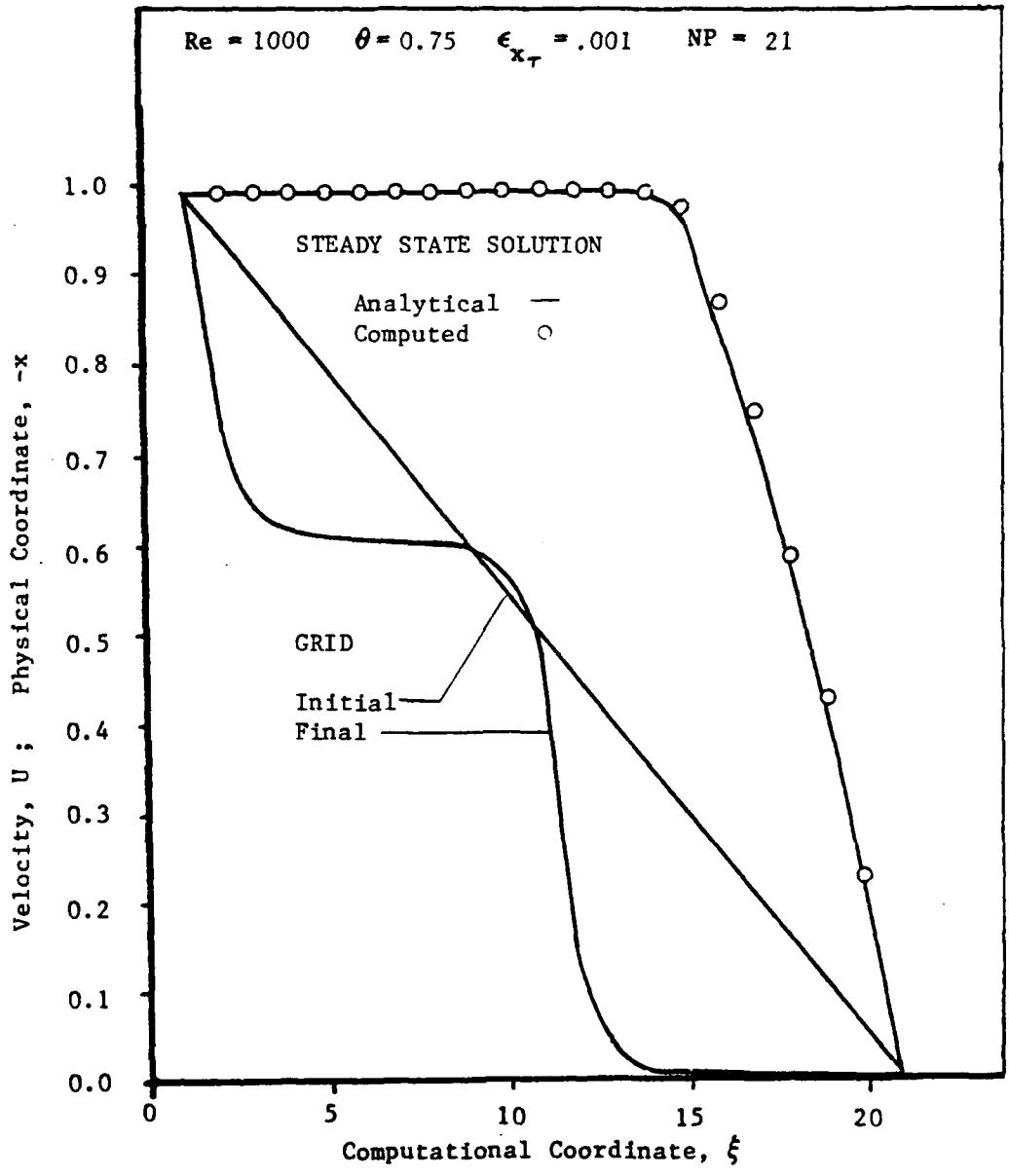


Fig 5 Velocity Profile & Grid Re = 1000 , NP = 21

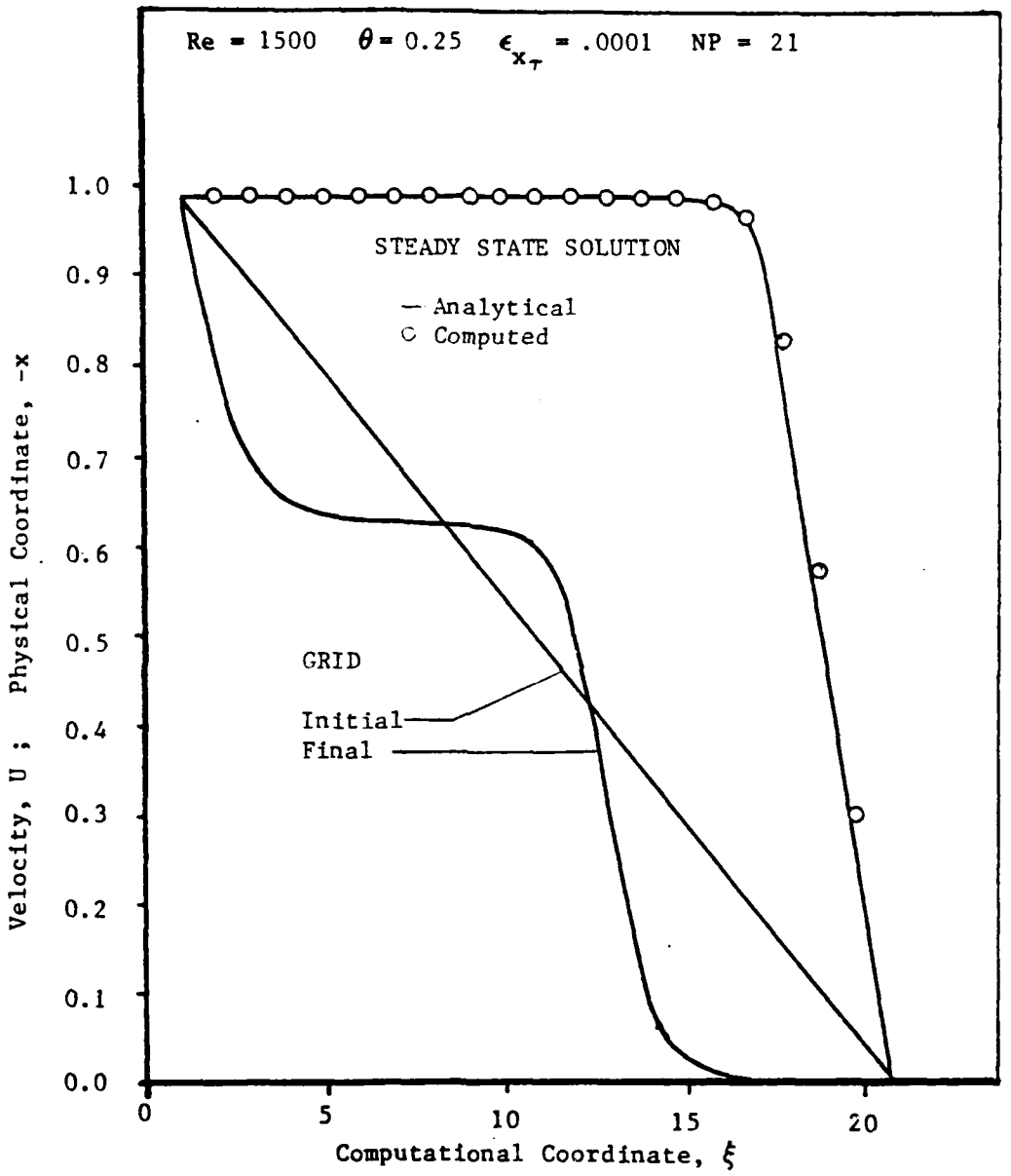


Fig 6 Velocity Profile & Grid Re = 1500 , NP = 21

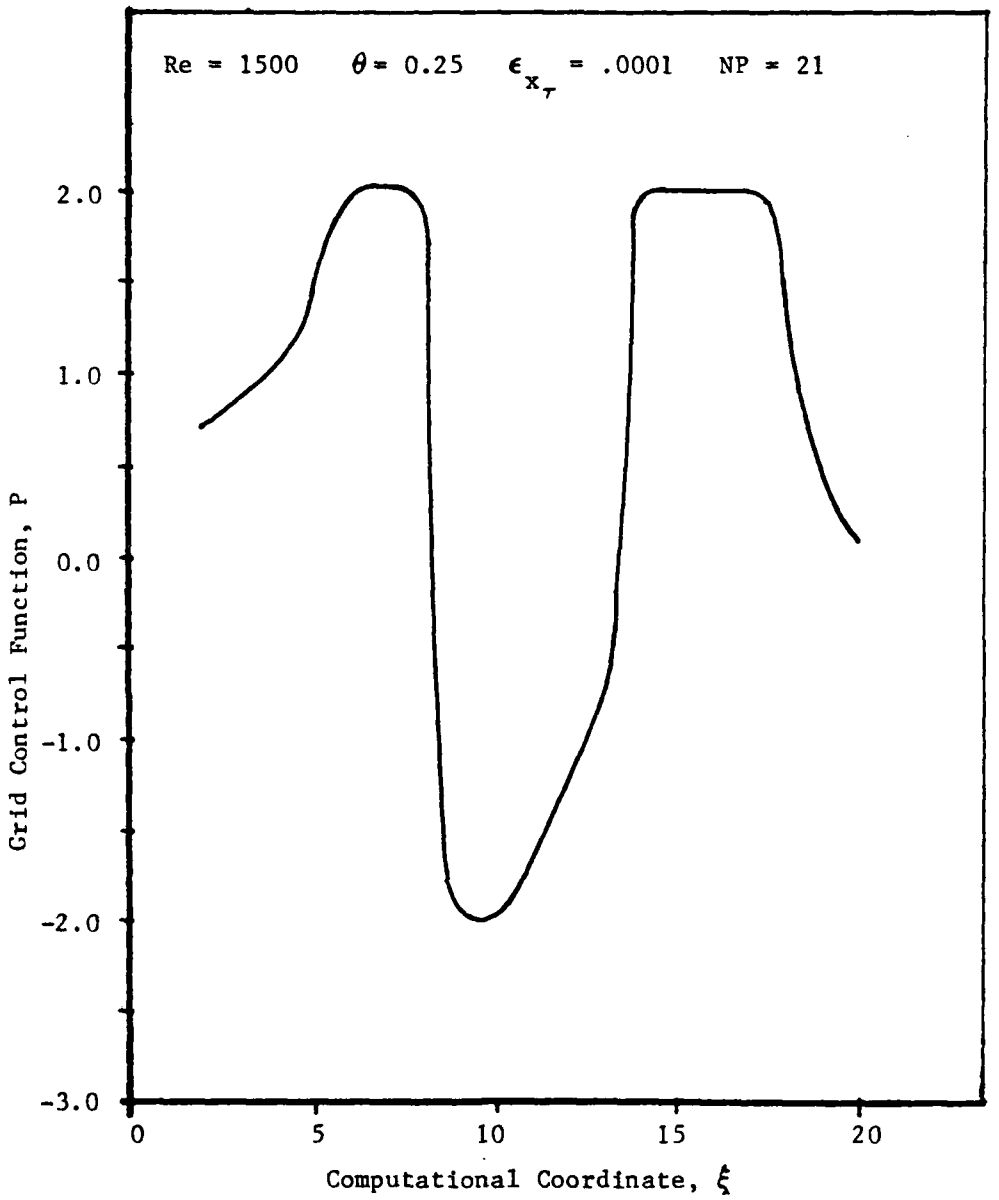


Fig 7 Grid Control Function Distribution, Re = 1500, NP = 21

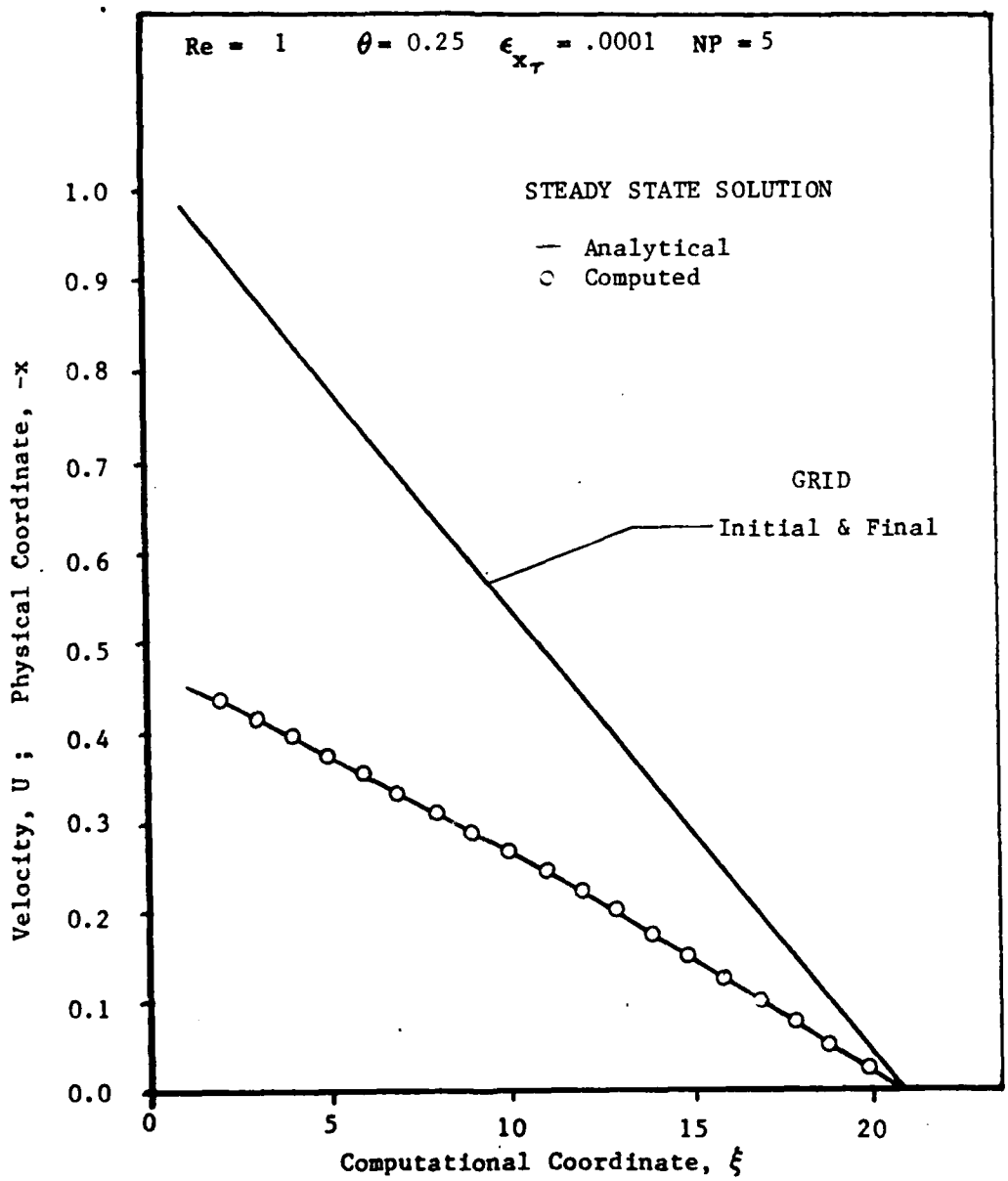


Fig 8 Velocity Profile & Grid Re = 1 , NP = 5

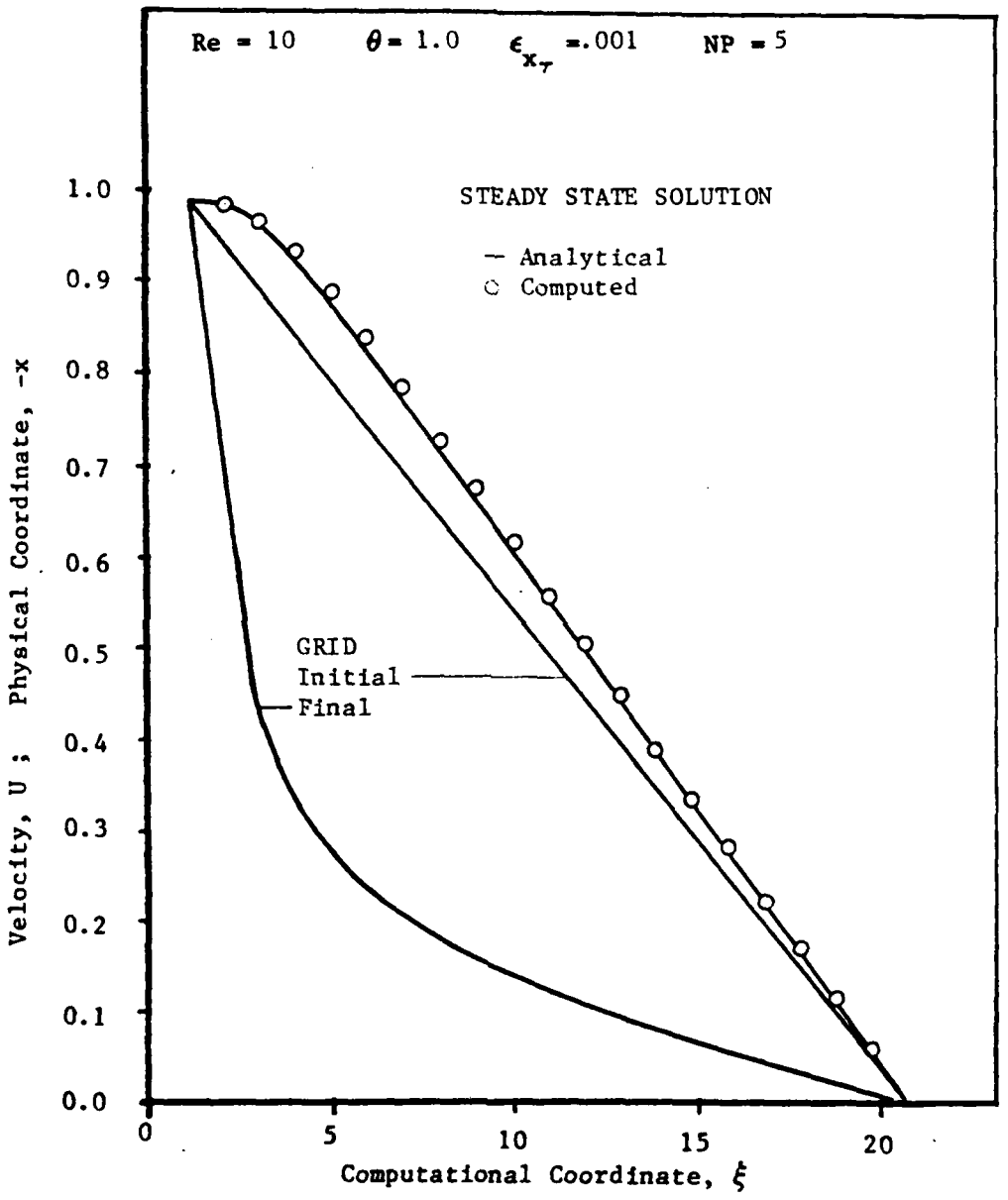


Fig 9 Velocity Profile & Grid Re = 10 , NP = 5

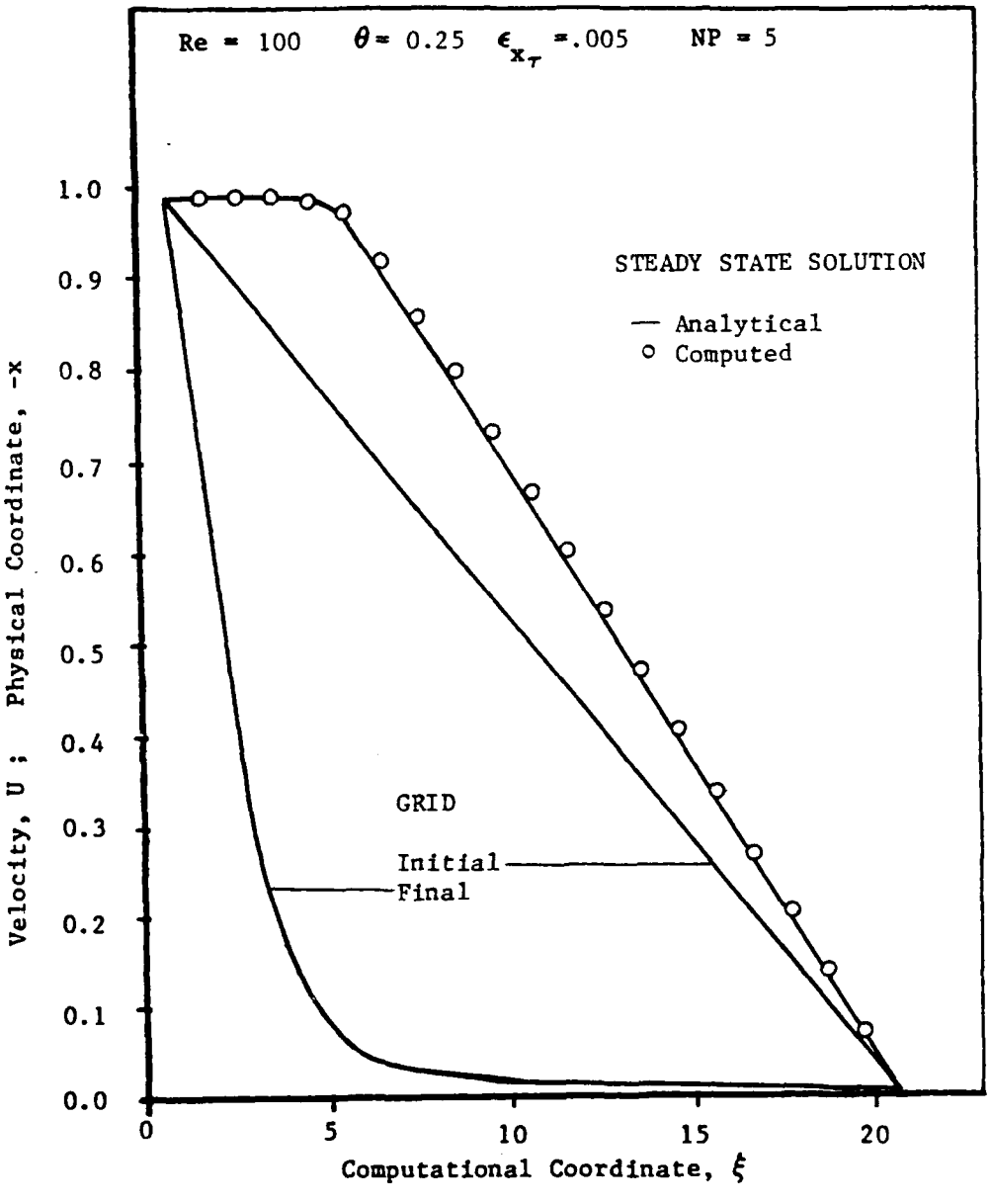


Fig 10 Velocity Profile & Grid Re = 100 , NP = 5

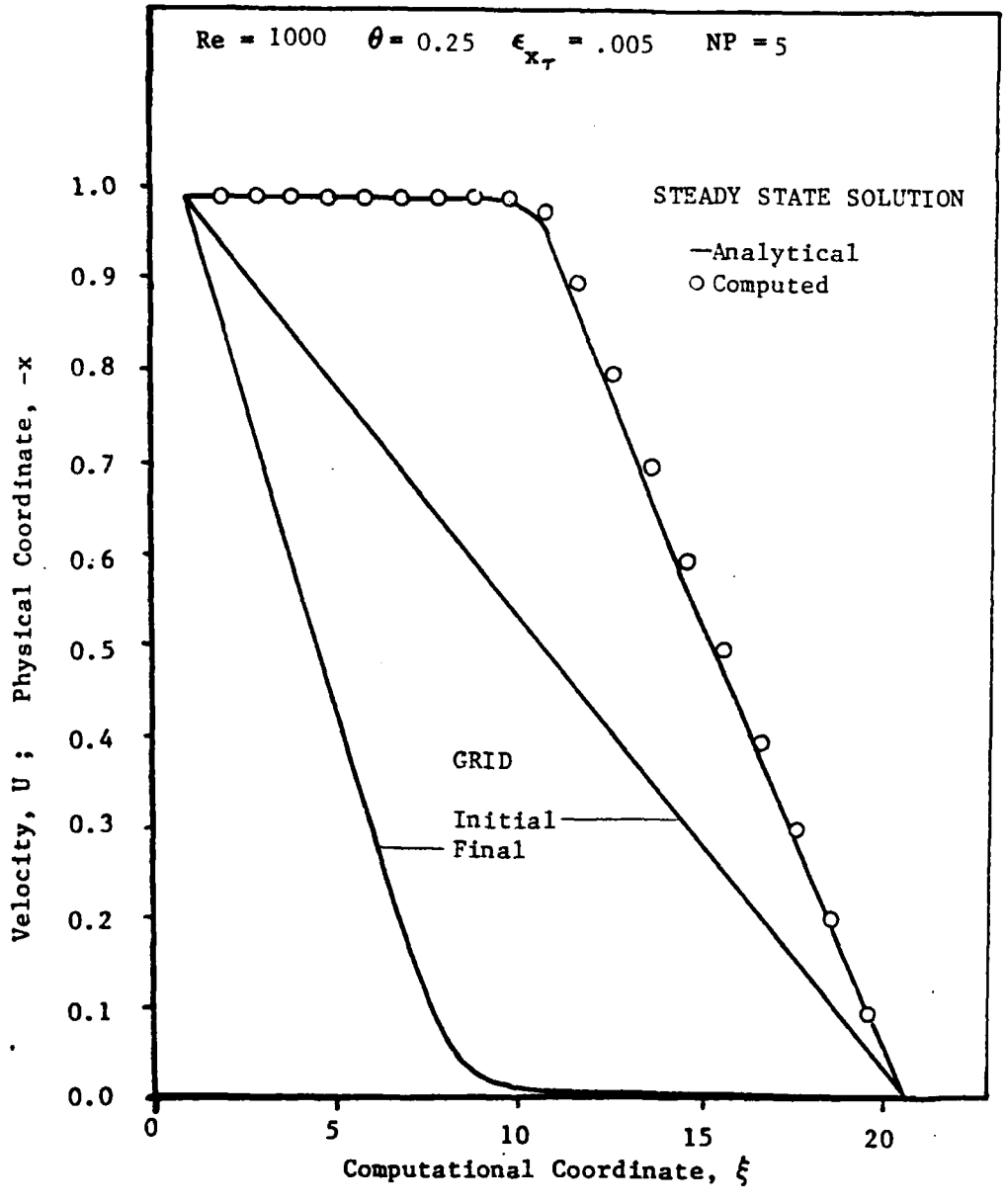


Fig 11 Velocity Profile & Grid Re = 1000, NP = 5

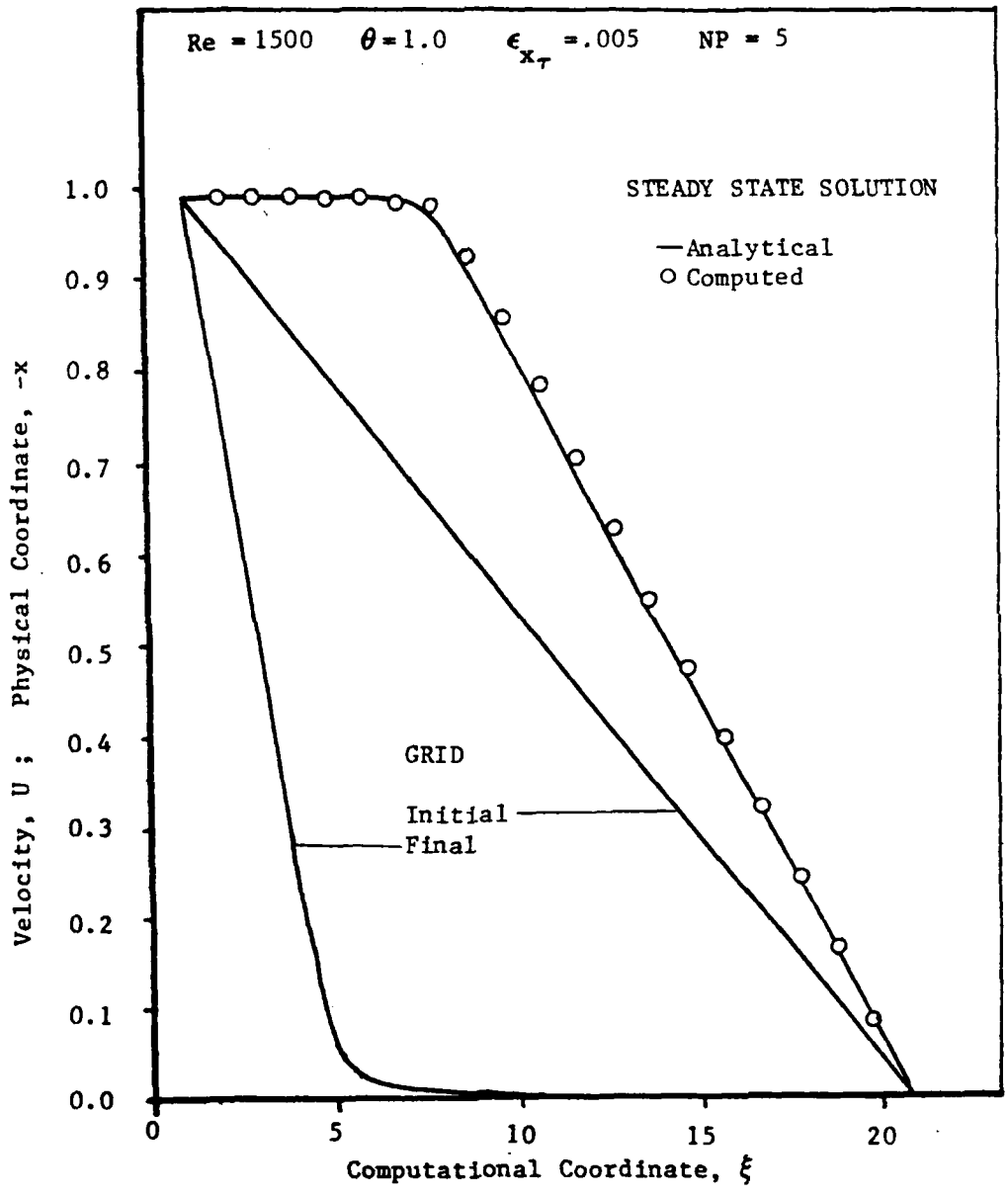


Fig 12 Velocity Profile & Grid Re = 1500 , NP = 5

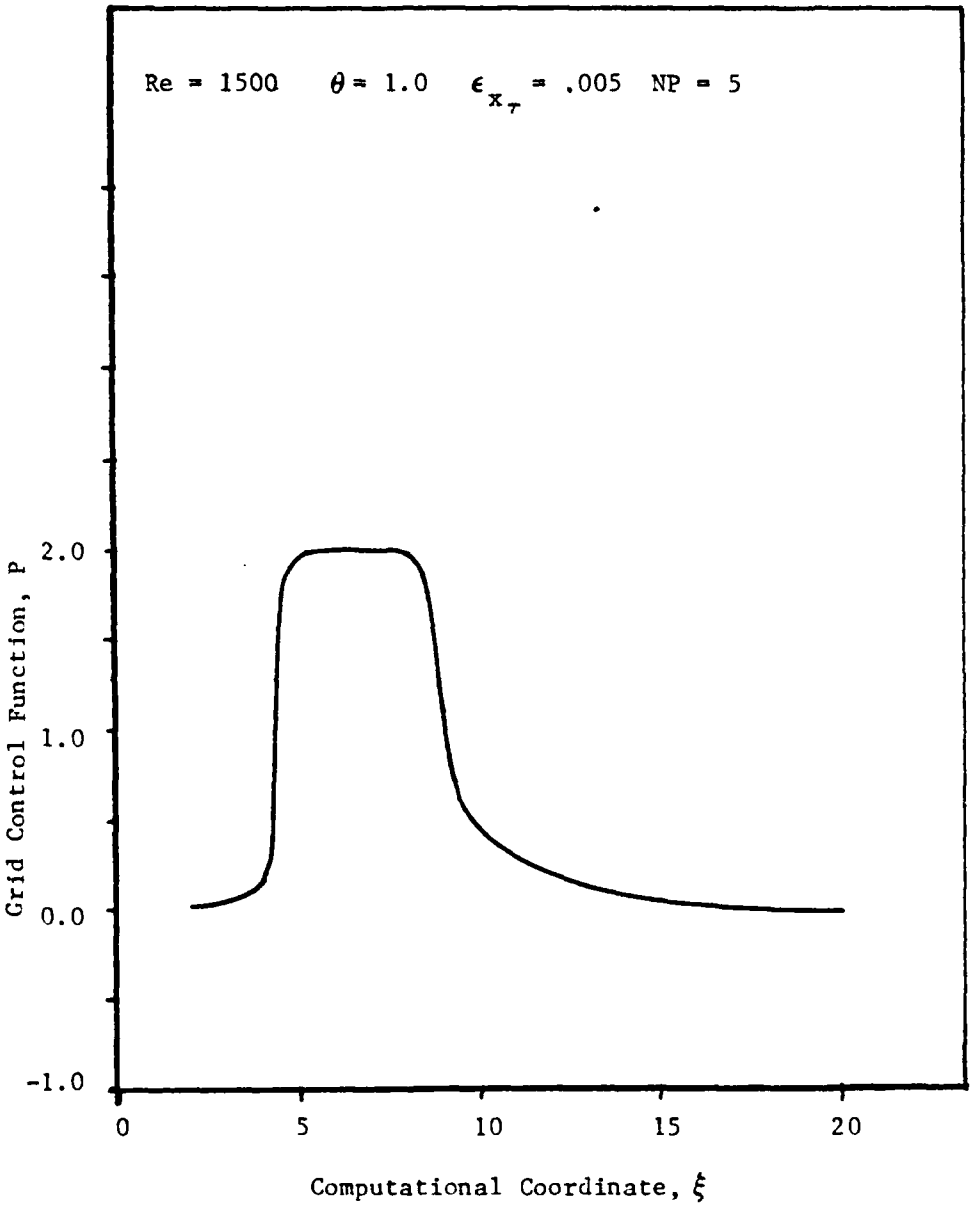


Fig 13 Grid Control Function Distribution, Re = 1500, NP = 5

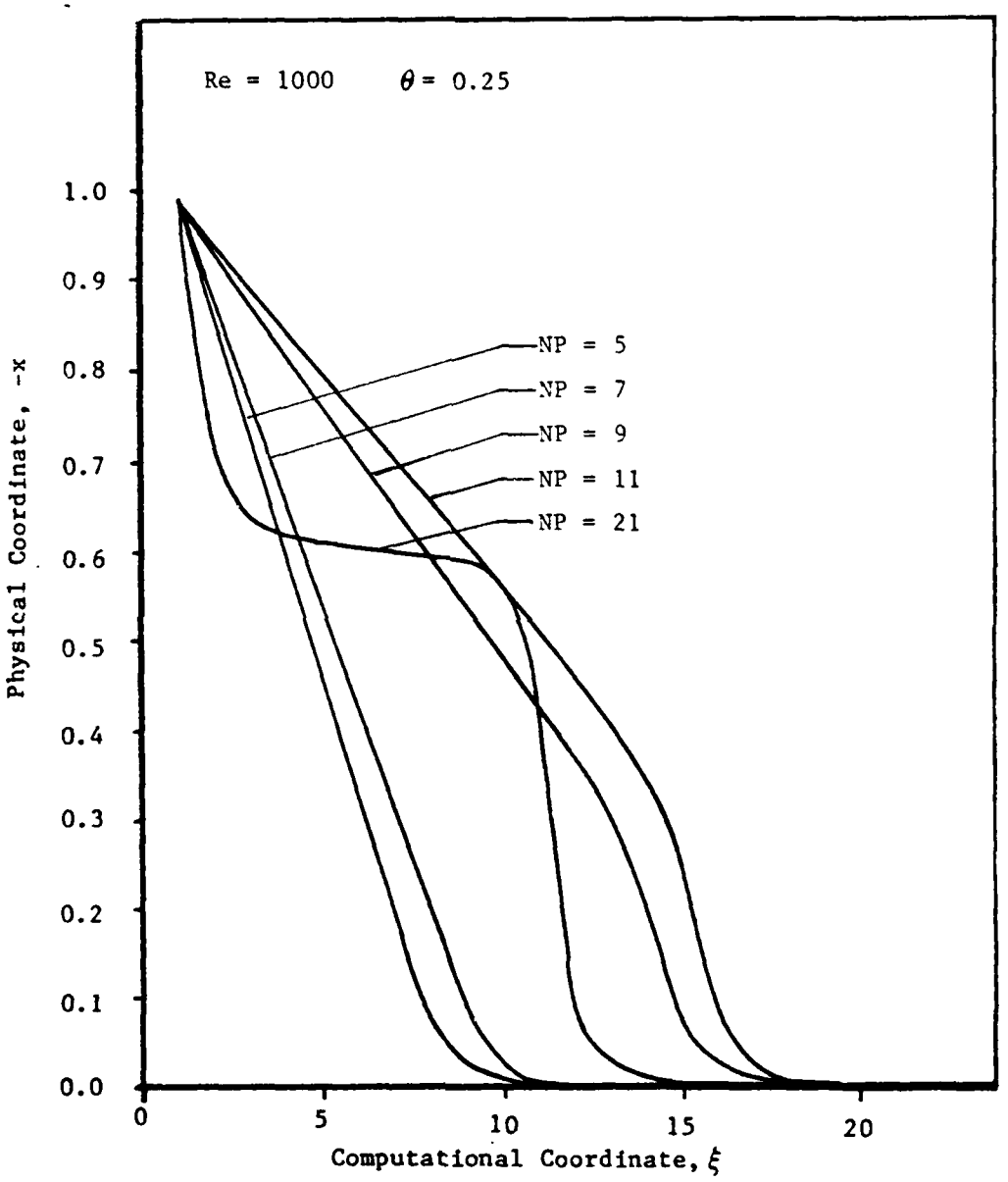


Fig 14 Grid Dependence on NP

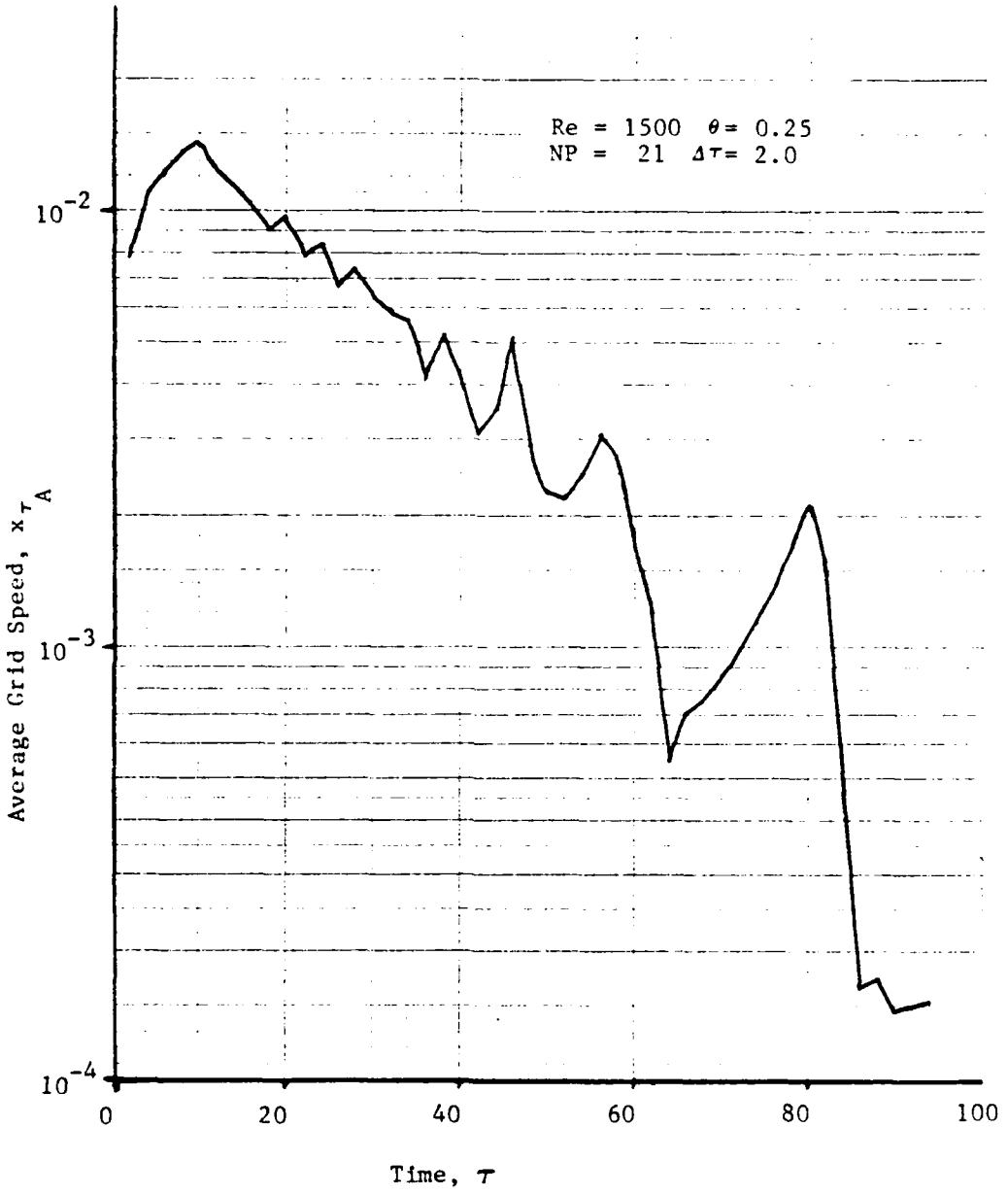


Fig 15 Grid Speed Time History

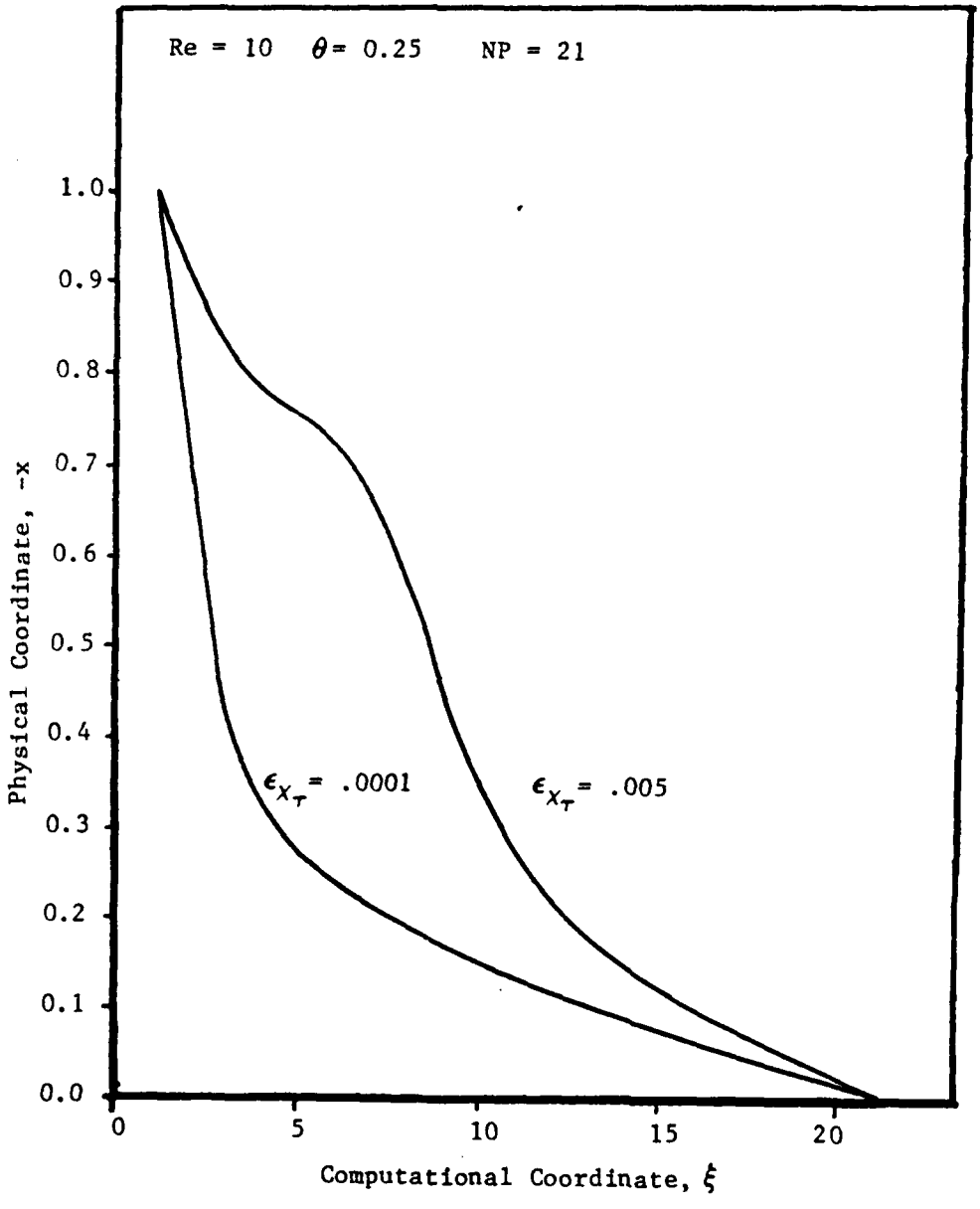


Fig 16 Grid Dependence on ϵ_{x_T}

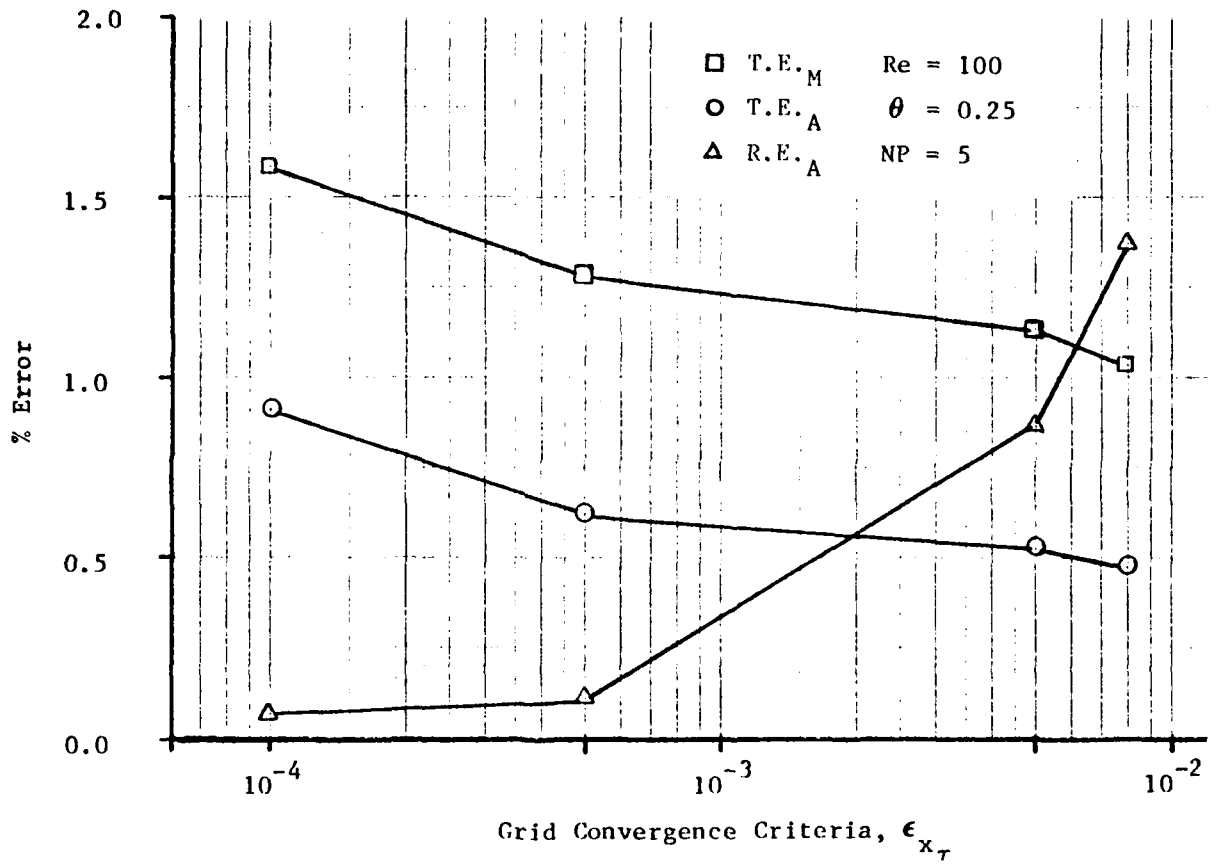


Fig 17 Solution Error Dependence on ϵ_{x_T}, Re = 100 , NP = 5

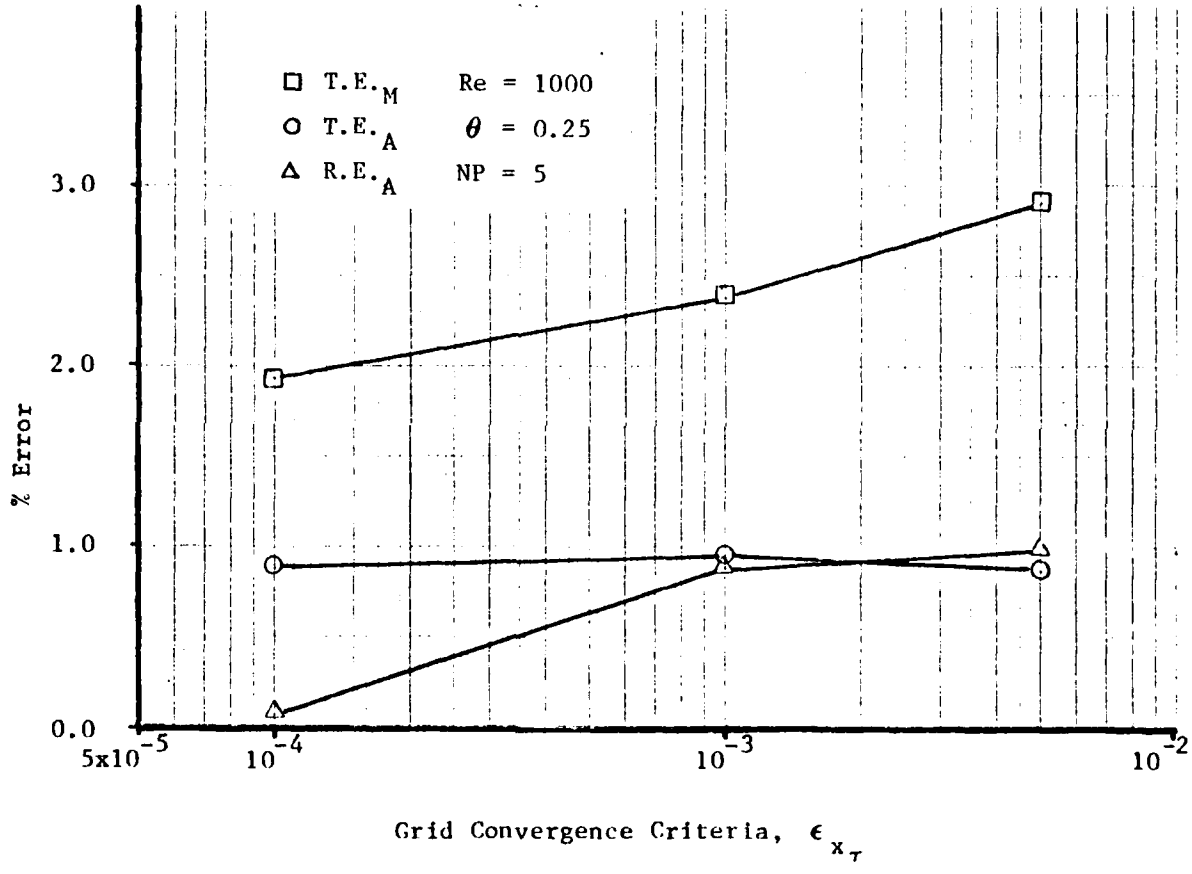


Fig 18 Solution Error Dependence on ϵ_{x_T}, Re = 100, NP = 5

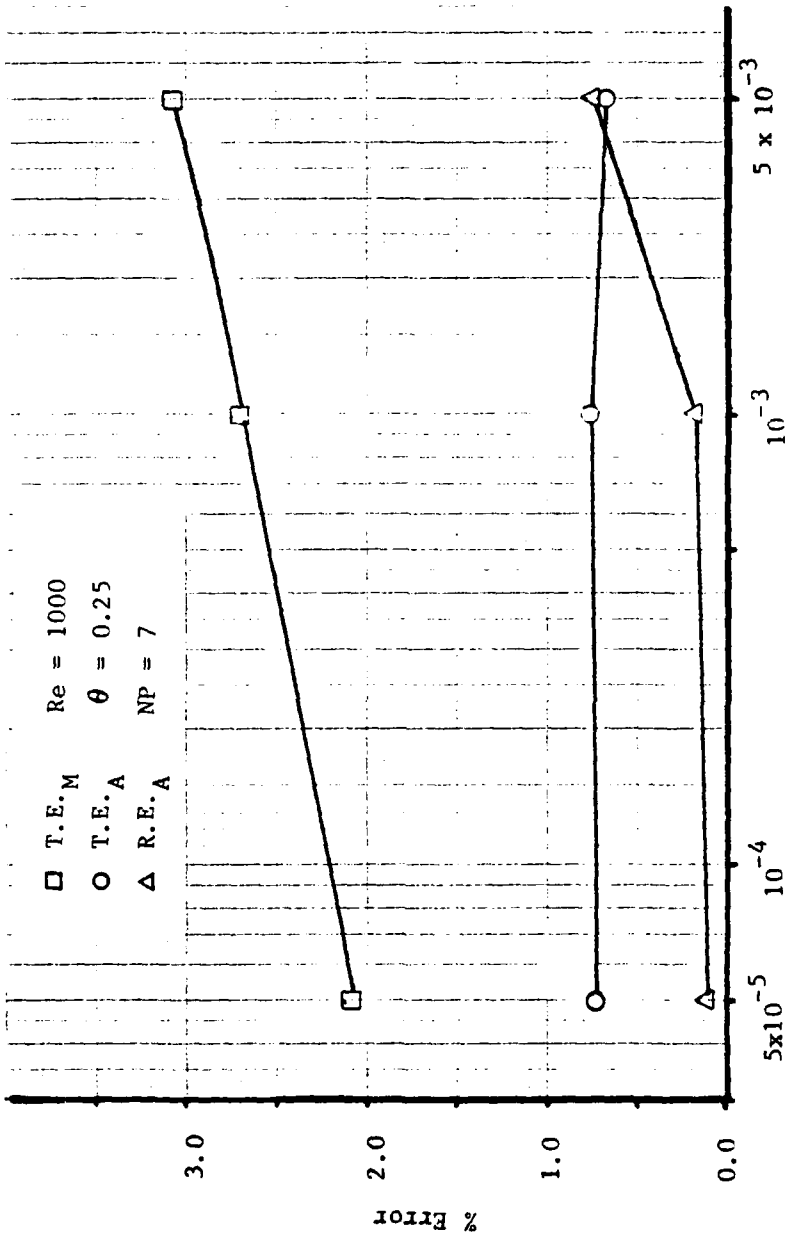


Fig 19 Solution Error Dependence on ϵ_{x_T}, Re = 1000, NP = 7

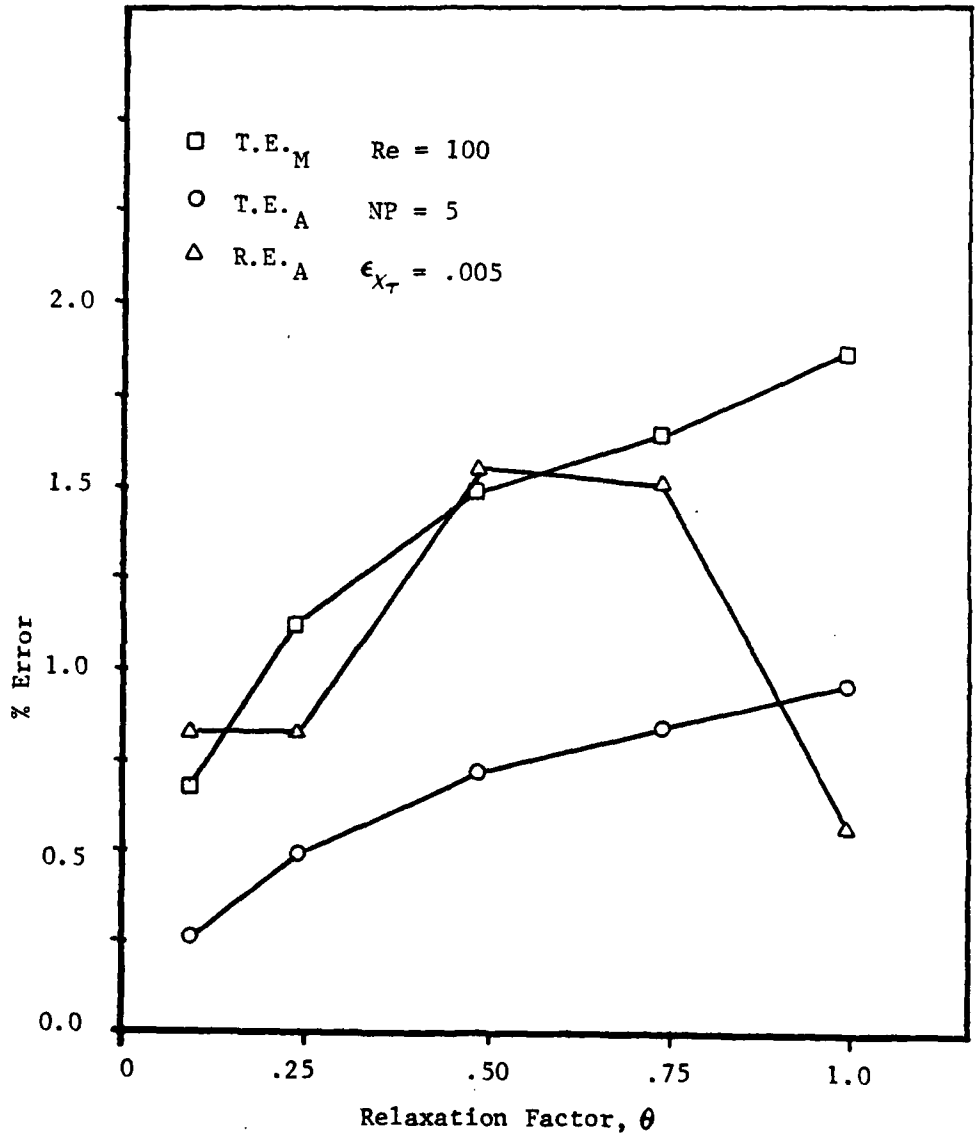


Fig 20 Solution Error Dependence on θ

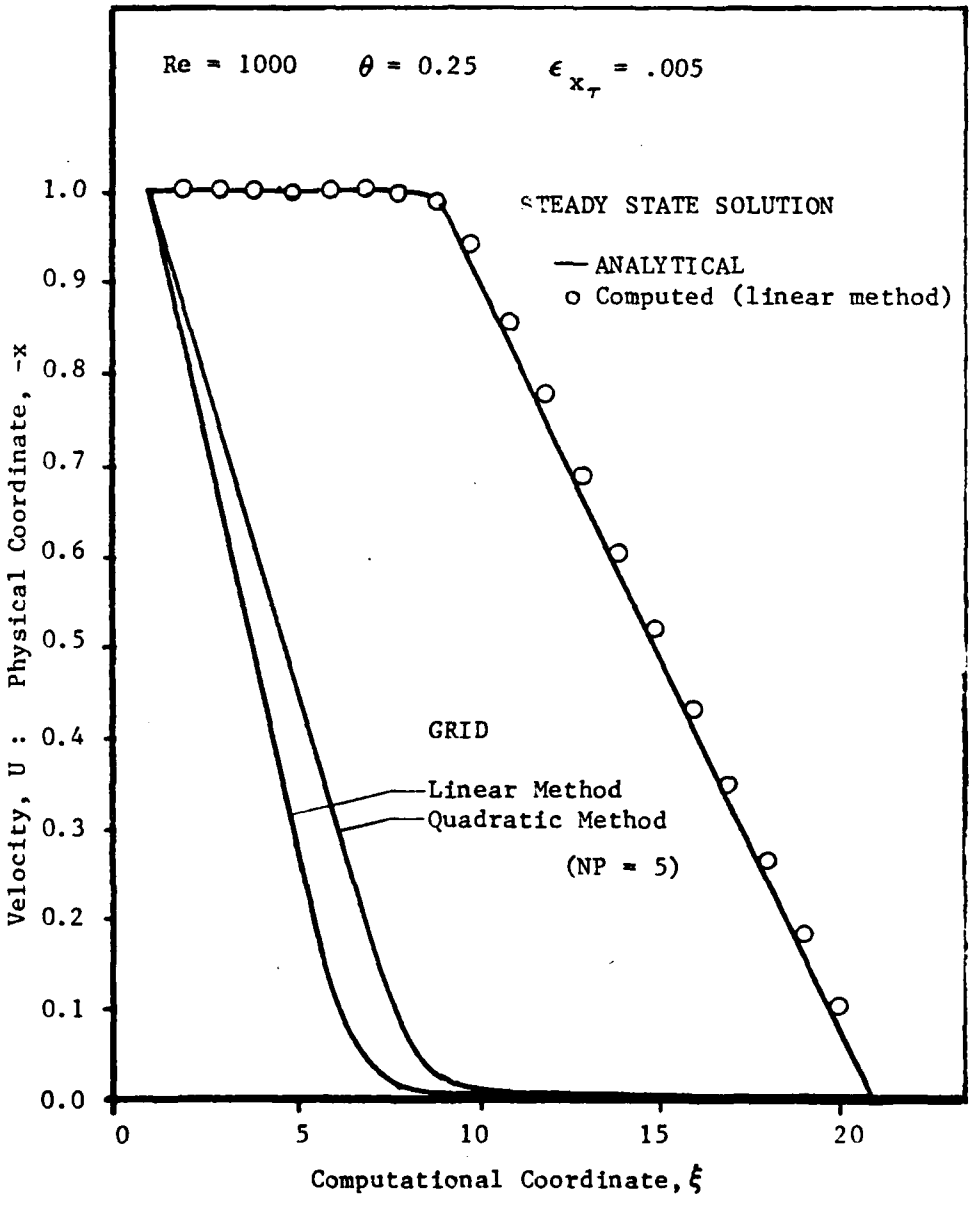


Fig 21 Grid Comparison with Linear Mothod

Appendix B

Program Listing

Appendix B

PROGRAM LISTING

```
      PROGRAM BURG15C(INPUT,OUTPUT,TAPE6,TAPE7,TAPE8)

C*****************************************************************
C     UNSTEADY SOLUTION OF THE 1-D BURGER'S EQUATION BY AN      *
C     OPTIMIZED SOR METHOD COUPLED WITH A GRID OPTIMIZATION     *
C     ROUTINE BASED ON A TRUNCATION ERROR ANALYSIS.  THE        *
C     PROCEDURE IS  NOT SELF-STARTING. THUS REQUIRING           *
C     AN INITIAL GUESS FOR THE GRID GENERATION CONTROL          *
C     PARAMETER, P.                                             *
C*****************************************************************

      COMMON /A/IMAX,U(51),ZI(51),X(51),P(51),DX(51),RE
      COMMON /B/KGRID,GRIDACC,XMIN,XMAX
      REAL DXX(51),XT(51),XN(51),UN(51),E(51),G(51),C(3)
      REAL LBC,RBC,XMIN,XMAX,ALPHA,BETA,GAMMA,DX2,A,B,USTAR,W,
     # UOLD,SUM,ERRMAX,ERRAVE,DIFF,GRIDACC,SOLNACC,DU,DUU,DELT,T,
     # XTAVE,XTACC,CPI,CPF,CPU,PI,ZZ(11),UU(11)
      INTEGER I,K,N,NT,KGRID,KSOLN
      PI=3.141592654

CC********* READ INPUT DATA ***************************************

      READ(7,*) IMAX,KGRID,KSOLN,LBC,RBC,XMIN,XMAX,NP
      READ(7,*) RE,GRIDACC,SOLNACC,DELT,NT,THETA,XTACC
      READ(7,*)(P(I),I=1,IMAX)
      CALL DATE(ADATE)
      CALL TIME(ATIME)
      WRITE(6,45)  ADATE,ATIME
      WRITE(6,50) IMAX,NT,RE
      WRITE(6,51) LBC,XMIN,RBC,XMAX
      WRITE(6,53) KSOLN,SOLNACC,KGRID,GRIDACC
      WRITE(6,54) DELT,XTACC,THETA,NP
      WRITE(8,200) ADATE,ATIME
      WRITE(8,205) IMAX,NT,RE
      WRITE(8,210) LBC,XMIN,RBC,XMAX
      WRITE(8,215) KSOLN,SOLNACC,KGRID,GRIDACC
      WRITE(8,210) DELT,XTACC,THETA,FLOAT(NP)

CC********* SET INITIAL CONDITIONS ********************************

      T=0.0
      N=0
      DO 10 I=1,IMAX
```

```
        ZI(I)=FLOAT(I)
        X(I)=XMIN+(I-1)*(XMAX-XMIN)/(IMAX-1)
        U(I)=1.0
 10   CONTINUE
      CALL USTART
      CALL UPSET('POLYNOMIAL',2.0)
      XTAVE=1.0

CC********** SET BOUNDARY CONDITIONS ******************************

      U(1)=-TANH(RE*X(1)*0.5)
      U(IMAX)=-TANH(RE*X(IMAX)*0.5)
      XT(1)=0.0
      XT(IMAX)=0.0

CC********* CALCULATE SOLUTION FOR EACH TIME STEP, N *************

      CALL SECOND(CPI.
 990  N=N+1
        T=T+DELT
        IF (N .GT. NT) GO TO 1000
        K=0
        DO 105 I=1,IMAX
          XN(I)=X(I)
 105      UN(I)=U(I)
        WRITE(6,65) T
        WRITE(8,210) T

CC********* CALCULATE GRID ***************************************

        IF (N  .LE. 5 ) GO TO 24
        IF (ABS(XTAVE) .LE. XTACC) GO TO 25
 24     IUPWND=0
        CALL GRIDSOR(IUPWND)
 25     CONTINUE

CC********* WRITE GRID DATA **************************************

        XTAVE=0.0
        WRITE(6,52)
        DO 20 I=2,IMAX-1
          DX(I)=0.5*(X(I+1)-X(I-1))
          DXX(I)=X(I+1)-2.0*X(I)+X(I-1)
          XT(I)=(X(I)-XN(I))/DELT
          XTAVE=XTAVE+XT(I)
          WRITE(6,58)I,X(I),DX(I),DXX(I),XT(I),P(I)
 20     CONTINUE
        XTAVE=XTAVE/(IMAX-2)
        WRITE(6,59)XTAVE
        IF (N .GE. NT-3) XTAVE=0.0

CC********* K IS THE SOR ITERATION LOOP INDEX *********************
```

```
        DO 900 K=1,KSOLN
          ERRMAX=0.0
          SUM=0.0

CC******** I IS THE SPACE LOOP INDEX ****************************

          DO 110 I=2,IMAX-1
            DX2=DX(I)*DX(I)
            ALPHA=1.0/(RE*DX2)
            BETA=(U(I)-XT(I) + DXX(I)*ALPHA)/DX(I)
C*********************************************************************
C      USE A FIRST ORDER DIFFERENCE NEXT TO THE  BOURNDARY     *
C      POINTS AND A SECOND ORDER DIFFERENCE  INTERIOR          *
C      TO THESE POINTS FOR THE UPWIND CONVECTIVE TERM          *
C      THE UPWIND VALUE IS BASED ON THE VALUE OF BETA          *
C*********************************************************************

            IF (BETA .GE. 0) THEN
              IF (I .EQ. 2 .OR. I .EQ. (IMAX-1)) THEN
                DU=-U(I-1)
                GAMMA = BETA + 2.0*ALPHA + 1.0/DELT
                A = ALPHA/GAMMA
                B = (ALPHA + BETA)/GAMMA
              ELSE
                DU = 0.5*U(I-2) - 2.0*U(I-1)
                GAMMA = 1.5*BETA + 2.0*ALPHA + 1.0/DELT
                A = ALPHA/GAMMA
                B = (ALPHA + BETA)/GAMMA
              END IF
            ELSE
              IF (I .EQ. 2 .OR. I .EQ. (IMAX-1)) THEN
                DU = U(I+1)
                GAMMA = -BETA + 2.0*ALPHA + 1.0/DELT
                A = (ALPHA - BETA)/GAMMA
                B = ALPHA/GAMMA
              ELSE
                DU = -0.5*U(I+2) + 2.0*U(I+1)
                GAMMA = -1.5*BETA + 2.0*ALPHA + 1.0/DELT
                A = (ALPHA - BETA)/GAMMA
                B = ALPHA/GAMMA
              END IF
            END IF

CC******** CALCULATE THE GAUSS-SEIDEL VALUE **********************

          USTAR=(UN(I)/DELT+ALPHA*(U(I+1)+U(I-1))-BETA*DU)/GAMMA

CC******** CALCULATE THE OPTIMUM RELAXATION FACTOR ***************

            CALL WOPT(A,B,IMAX,W)
            IF(N .EQ. 1 .AND. K .EQ. 1) W=1.0
```

```
C**** CALCULATE SOLUTION & ERROR BETWEEN SOR ITERATIONS ********

            UOLD = U(I)
            U(I) = U(I) + W*(USTAR - U(I))
            DIFF = ABS(U(I) - UOLD)
            SUM = SUM + DIFF
 110      CONTINUE

CC******** CHECK FOR ITERATION CONVERGENCE **********************

          ERRAVE = SUM/IMAX
CC         IF (ERRAVE .LE. SOLNACC) GO TO 910
 900      CONTINUE
 910      WRITE(6,79)

C***** CALCULATE NEW GRID GENERATION CONTROL FUNCTION P ********

        IF (N .LE. 5) GO TO 34
        IF (ABS(XTAVE) .LE. XTACC) GO TO 35
 34     CALL NEWP(THETA,C,NP,G)
 35     CONTINUE

CC******** CALCULATE ANALYTIC STEADY STATE SOLUTION *************

        WRITE(6,87)
        ERR1=0.0
        SUM1=0.0
        ERR2=0.0
        SUM2=0.0
        ERR3=0.0
        SUM3=0.0
        DO 120 I=1,IMAX
          E(I)=-TANH(RE*X(I)*0.5)
          DIF1=ABS(U(I)-E(I))
          SUM1=SUM1+DIF1
          DIF2=ABS(U(I)-G(I))
          SUM2=SUM2+DIF2
          DIF3=ABS(G(I)-E(I))
          SUM3=SUM3+DIF3
          WRITE(6,58)I,X(I),U(I),E(I),DIF1,G(I)
          WRITE(8,225)X(I),XT(I),P(I),U(I),G(I),E(I)
 120    CONTINUE
        ERRAV1=SUM1/IMAX
        ERRAV2=SUM2/IMAX
        ERRAV3=SUM3/IMAX
        WRITE(6,80)K-1,ERRAV1,ERRAV2,ERRAV3
        WRITE(8,230)K-1,ERRAV1,ERRAV2,ERRAV3

C***** CALCULATE AVERAGE ERROR BETWEEN SOR ITERATIONS **********

        SUM=0.0
```

```
      DO 180 I=1,IMAX
        DIFF=ABS(U(I)-UN(I))
        SUM=SUM+DIFF
180   CONTINUE
      ERRAVE=SUM/IMAX
      WRITE(6,82)ERRAVE
      WRITE(8,210)ERRAVE
    IF (ERRAVE .GE. SOLNACC) GO TO 990

    CALL SECOND(CPF)
    CPU=CPF-CPI
    WRITE(6,95)CPU
    WRITE(8,235)CPU
    CALL UEND
    STOP

200 FORMAT(T5,A10,5X,A10)

205 FORMAT(T5,2I5,E12.4)

210 FORMAT(T5,4E12.4)

215 FORMAT(T5,I5,E12.4,I5,E12.4)

225 FORMAT(6E11.4)

230 FORMAT(T5,I5,3E12.4)

235 FORMAT(T5,E12.4)

45  FORMAT('1',/,T5,A10,2X,A10)

50  FORMAT(/,T10,'1-D BURGERS EQ. SOLN USING OPTIMIZED SOR METHOD'
   #,' & OPTIMUM GRID',//,T25,'**********INPUT DATA**********',/3X,
   #'# OF GRID PTS =',I3,3X,'# OF TIME STEPS ',I3,3X,'RE # =',E9.3)

51  FORMAT(3X,'U =',F5.2,' @ X =',F5.2,3X,'U =',F5.2,' @ X =',F5.2)

53  FORMAT(3X,'# OF SOLN ITER. =',I3,3X,'SOLN ACC CK =',
   #E9.3,/,3X,'# OF GRID ITER. =',I3,3X,'GRID ACC CK =',E9.3)

54  FORMAT(3X,'DELTA TIME =',F5.2,3X,'GRID SPEED CK =',E9.3,/,
   #3X,'NEW P RELAXATION FACTOR:',F4.2,3X,'# OF PTS IN ',
   #'CURVE FIT:',I3,/)

52  FORMAT(/,T25,'GRID CALCULATION',/,T4,'ZI',T14,'X',T25,
   #'DX',T38,'DXX',T50,'XT',T62,'P',/)

58  FORMAT(I5,8(F12.7))

59  FORMAT(/,T5,'AVERAGE VALUE OF GRID  SPEED =',F12.7)
```

```
 65   FORMAT(//,T30,'TIME =',F6.3,/)

 79   FORMAT(//)

 80   FORMAT(T5,'SOR ITERATION #',I5.5X,'AVERAGE ERROR ANALYSIS',/,
     #T30,'COMPUTED & ANALYTICAL :',F13.8,/,T30,'COMPUTED & FIT :',
     #8X,F12.8,/,T30,'ANALYTICAL & FIT :',5X,F13.8)

 82   FORMAT(T5,'AVE ERROR BETWEEN TIME STEPS =',F12.8)

 87   FORMAT(/,T30,'*****  ANALYTIC SOLUTION  *****',//
     #T4,'ZI',T14,'X',T25,'U',T35,'U EXACT',T47,'DIFF',T60,'FIT',/)

 90   FORMAT(/,T20,'LEAST SQUARES FIT TO SLUTION',/,T5,'U(ZI) = ',
     #F10.6,' * ZI**2  +  ',F10.6,' * ZI  +  ',F10.6,/)

 95   FORMAT(/,T5,'TOTAL CPU TIME :', E17.7)

 98   FORMAT(/,T5,'MAX ITERATIONS EXCEEDED!!!!!',/)

1000 WRITE(6,98)
     CALL SECOND(CPF)
     CPU=CPF-CPI
     WRITE(6,95)CPU
     WRITE(8,235)CPU

     STOP
     END



     SUBROUTINE WOPT(A,B,IMAX,W)

     PI=3.141592654
     PJ=2.0*SQRT(A*B)*0.99
     W=2.0/(1.0+SQRT(1.0-PJ**2))
     RETURN
     END



     SUBROUTINE GRIDSOR(IUPWND )

C*********************************************************************
C   THIS SUBROUTINE CALCULATES THE SOLUTION OF THE 1-D               *
C   GRID GENERATION EQUATION BY THE THOMAS ALGORITHM.                *
C   IF IUPWND = 1, THEN IT USES 2ND ORDER UPWIND DIFFERENCES         *
C   FOR THE CONVECTIVE TERM. IF IUPWND = 0, IT USES  A 2ND           *
C   ORDER CENTRAL DIFFERENCE FOR THE CONVECTIVE TERM                 *
C*********************************************************************
```

```
      COMMON /A/IMAX,U(51),ZI(51),X(51),P(51),DUMMY(51),RE
      COMMON /B/KGRID,GRIDACC,XMIN,XMAX
      WRITE(6,56)
      X(1)=XMIN
      X(IMAX)=XMAX
      K = 0
CC******** K IS THE SOR ITERATION LOOP INDEX ********************

 999      K = K+1
          IF (K .GE. KGRID) GO TO 200
          ERRMAX=0.0
          SUM=0.0

CC******** I IS THE SPACE LOOP INDEX ****************************

          DO 110 I=2,IMAX-1
            ALPHA=1.0
            BETA=-P(I)
C*****************************************************************
C       USE A FIRST ORDER DIFFERENCE NEXT TO THE  BOURNDARY      *
C       POINTS AND A SECOND ORDER DIFFERENCE  INTERIOR           *
C       TO THESE POINTS FOR THE UPWIND CONVECTIVE TERM           *
C       THE UPWIND VALUE IS BASED ON THE VALUE OF BETA           *
C*****************************************************************

            IF (BETA .GE. 0) THEN
              IF (I .EQ. 2 .OR. I .EQ. (IMAX-1)) THEN
                DX=-X(I-1)
                GAMMA = BETA + 2.0*ALPHA
                A = ALPHA/GAMMA
                B = (ALPHA + BETA)/GAMMA
              ELSE
                DX = 0.5*X(I-2) - 2.0*X(I-1)
                GAMMA = 1.5*BETA + 2.0*ALPHA
                A = ALPHA/GAMMA
                B = (ALPHA + BETA)/GAMMA
              END IF
            ELSE
              IF (I .EQ. 2 .OR. I .EQ. (IMAX-1)) THEN
                DX = X(I+1)
                GAMMA = -BETA + 2.0*ALPHA
                A = (ALPHA - BETA)/GAMMA
                B = ALPHA/GAMMA
              ELSE
                DX = -0.5*X(I+2) + 2.0*X(I+1)
                GAMMA = -1.5*BETA + 2.0*ALPHA
                A = (ALPHA - BETA)/GAMMA
                B = ALPHA/GAMMA
              END IF
            END IF
```

```
CC******** CALCULATE THE GAUSS-SEIDEL VALUE ********************

            XSTAR = (ALPHA*(X(I+1)+X(I-1))-BETA*DX)/GAMMA

CC******** CALCULATE THE OPTIMUM RELAXATION FACTOR *************

            CALL WOPT(A,B,IMAX,W)
            IF(K .EQ. 1) W=1.0

C***** CALCULATE SOLUTION & ERROR BETWEEN SOR ITERATIONS *******

            XOLD = X(I)
            X(I) = X(I) + W*(XSTAR - X(I))
            DIFF = ABS(X(I) - XOLD)
            IF (DIFF .GE. ERRMAX) THEN
              ERRMAX = DIFF
              II = I
            END IF
            SUM = SUM + DIFF
 110      CONTINUE

C********** CHECK FOR ITERATION CONVERGENCE **********************

          ERRAVE = SUM/IMAX
          IF (ERRAVE .GE. GRIDACC) GO TO 999

 200  WRITE(6,80)K,ERRMAX,II,ERRAVE

 56   FORMAT(/T15,'SOR GRID CALCULATION ',/,T5,'USING A 2ND ORDER',
     #' UPWIND DIFFERENCE FOR THE CONVECTIVE TERM')

 80   FORMAT(T5,'ITERATION #',I5,5X,'MAX ERROR =',F10.5,'  AT',
     #I4,/, 5X,'AVERAGE ERROR =',F10.5)

      RETURN
      END



      SUBROUTINE NEWP(THETA,C,NP,G)

C*****************************************************************
C    THIS SUBROUTINE CACULATES THE NEW GRID GENERATION CONTROL *
C    FUNCTION P.   IT IS BASED ON A GLOBAL TRUNCATION ERROR    *
C    ANALYSIS FOR THE FLOW SOLUTION IN THE TRANSFORMED PLANE   *
C*****************************************************************

      COMMON /A/IMAX,U(51),ZI(51),X(51),P(51),DX(51),RE
      REAL C(3),ZZ(11),UU(11),G(51)
      NN=(NP-1)/2
      WRITE(6,15)
```

```
      DO 7 J=1,NP
        ZZ(J)=ZI(J)
        UU(J)=U(J)
 7    CONTINUE
      CALL ULSTSQ(ZZ,UU,FLOAT(NP),C)
      A2=C(3)
      A1=C(2)
      A0=C(1)
CC    WRITE(6,27)(ZZ(J),J=1,NP)
CC    WRITE(6,28)(UU(J),J=1,NP)
CC    WRITE(6,29) A2,A1,A0
     DO 5 I=2,NN+1
      G(I) = A2*I**2 + A1*I + A0
      DU=0.5*(U(I+1)-U(I-1))
      DUU=U(I+1)-2.0*U(I)+U(I-1)
      DUSIGN=SIGN(1.0,DU)
      IF(ABS(DU) .LE. .0000001) DU=.0000001*DUSIGN
      DEN=2.0*A2*I + A1
      DENSN=SIGN(1.0,DEN)
      IF (ABS(DEN) .LE. 0.0001 ) DEN=.0001*DENSN
      POLD=P(I)
      P(I)=POLD + THETA*DUU/DU - THETA*2.0*A2/DEN
      PMAX = U(I)*RE*DX(I)
      IF (P(I) .GT.  PMAX) P(I)=PMAX
      IF (P(I) .GT. 2.0) P(I)=2.0
      IF (P(I) .LT. -2.0) P(I)=-2.0
      WRITE(6,16)I,DU,DUU,DUU/DU,POLD,P(I)
 5    CONTINUE

     DO 10 I=2+NN,IMAX-NN-1
      DO 20 J=1,NP
        ZZ(J) = ZI(I+J-NN-1)
        UU(J) =  U(I+J-NN-1)
 20   CONTINUE
      CALL ULSTSQ(ZZ,UU,FLOAT(NP),C)
      A2=C(3)
      A1=C(2)
      A0=C(1)
CC    WRITE(6,27)(ZZ(J),J=1,NP)
CC    WRITE(6,28)(UU(J),J=1,NP)
CC    WRITE(6,29) A2,A1,A0
      G(I) = A2*I**2 + A1*I + A0
      DU=0.5*(U(I+1)-U(I-1))
      DUU=U(I+1)-2.0*U(I)+U(I-1)
      DUSIGN=SIGN(1.0,DU)
      IF(ABS(DU) .LE. .0000001) DU=.0000001*DUSIGN
      DEN=2.0*A2*I + A1
      DENSN=SIGN(1.0,DEN)
      IF (ABS(DEN) .LE. 0.0001 ) DEN=.0001*DENSN
      POLD=P(I)
      P(I)=POLD + THETA*DUU/DU - THETA*2.0*A2/DEN
      PMAX = U(I)*RE*DX(I)
```

```
        IF (P(I) .GT.  PMAX) P(I)=PMAX
        IF (P(I) .GT. 2.0) P(I)=2.0
        IF (P(I) .LT. -2.0) P(I)=-2.0
        WRITE(6,16)I,DU,DUU,DUU/DU,POLD,P(I)
 10   CONTINUE

        DO 30 J=NP,1,-1
          ZZ(J) = ZI(IMAX+J-NP)
          UU(J) =  U(IMAX+J-NP)
 30     CONTINUE
        CALL ULSTSQ(ZZ,UU,FLOAT(NP),C)
        A2=C(3)
        A1=C(2)
        A0=C(1)
CC      WRITE(6,27)(ZZ(J),J=1,NP)
CC      WRITE(6,28)(UU(J),J=1,NP)
CC      WRITE(6,29) A2,A1,A0
      DO 25 I=IMAX-NN,IMAX-1
        G(I) = A2*I**2 + A1*I + A0
        DU=0.5*(U(I+1)-U(I-1))
        DUU=U(I+1)-2.0*U(I)+U(I-1)
        DUSIGN=SIGN(1.0,DU)
        IF(ABS(DU) .LE. .0000001) DU=.0000001*DUSIGN
        DEN=2.0*A2*I + A1
        DENSN=SIGN(1.0,DEN)
        IF (ABS(DEN) .LE. 0.0001 ) DEN=.0001*DENSN
        POLD=P(I)
        P(I)=POLD + THETA*DUU/DU - THETA*2.0*A2/DEN
        PMAX = U(I)*RE*DX(I)
        IF (P(I) .GT.  PMAX) P(I)=PMAX
        IF (P(I) .GT. 2.0) P(I)=2.0
        IF (P(I) .LT. -2.0) P(I)=-2.0
        WRITE(6,16)I,DU,DUU,DUU/DU,POLD,P(I)
 25   CONTINUE
      P(1)=0.0
      G(1)=U(1)
      G(IMAX)=U(IMAX)
      P(IMAX)=0.0
 15   FORMAT(/,T25,'********* NEW P CALCULATION ******',//
     #T4,'ZI',T14,'DU',T26,'DUU',T36,'DUU/DU',
     #T48,'POLD',T60,'PNEW',/)

 16   FORMAT(I5,8(3X,F9.5))

 27   FORMAT(T2,'ZZ(J): ',6F10.5)

 28   FORMAT(  T2,'UU(J): ',6F10.5)

 29   FORMAT(T2,'LST SQ FIT  UU(ZZ) = ',
     #F8.5,' * ZI**2  +  ',F8.5,' * ZI  +  ',F8.5,)

      RETURN
```

Bibliography

1. Thompson, J. F., "A Survey of Grid Generation Tecniques in Computational Fluid Dynamics," AIAA Paper No. 83-0447, 1-36, (January 1983).

2. Ghia, K. N., Ghia, U., and Shin, C. T., "Adaptive Grid Generation for Flows with Local High Gradient Regions," Advances in Grid Generation, edited by K. N. Ghia and U. Ghia. ASME FED, 5: 35-47 (1983).

3. Thompson, J. F., Thames, F. C., and Mastin, C. W., "Automated Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing Any Number of Arbitrary Two-Dimensional Bodies," Journal of Computational Physics, 15: 299-319 (July 1974).

4. Ghia, U., Hodge, J.K., and Hankey, W.L., "An Optimization Study for Generating Surface-Oriented Coordinates for Arbitrary Bodies in High Reynolds Number Flows," AFFDL-TR-77-117, (December 1977).

5. Pierson, B. L., and Kutler, P., "Optimal Nodal Point Distribution for Improved Accuracy in Computational Fluid Dynamics," AIAA Journal, 18: 49-54 (January 1980).

6. Saltzman, J. and Brackbill, J., "Applications and Generalizations of Variational Methods for Generationg Adaptive Meshes," Numerical Grid Generation, edited by J. F. Thompson. New York: North Holland, 1982.

7. Dwyer, H. A., Kee, R. J., Sanders, B. R., "Adaptive Grid Method for Problems in Fluid Mechanics, and Heat Transfer," AIAA Journal, 18: 1205-1212 (October 1980).

8. Freeman, L. M., The Use of an Adaptive Grid in a Solution of the Navier-Stokes Equations for Incompressible Flows, Ph.D. Dissertation, Mississippi State University, December 1982.

9. Anderson, D. A., and Rai, M. M., "The Use of Solution Adaptive Grids in Solving Partial Differential Equations," Numerical Grid Generation, Edited by J. F. Thompson. New York: North Holland, 1982.

10. Thompson, J. F., and Mastin, C. W., "Grid Generation Using Differential Systems Techniques," Numerical Grid Generation Techniques, edited by R. E. Smith. NASA Conference Publication 2166, NASA Langley Research

Center, 1980.

11. Rai, M. M., and Anderson, D. A. "Grid Evolution in Time Asymptotic Problems," Numerical Grid Generation Techniques, edited by R. E. Smith. NASA Conference Publication 2166, NASA Langley Research Center, 1980.

12. Hodge, J. K., Stone, A. L., and Miller, T. E., "Numerical Solution for Airfoils near Stall in Optimized Boundary-Fitted Curvilinear Coordinates," AIAA Paper 78-284, AIAA 16th Aerospace Sciences Meeting, Huntsville, Al., (January 1978).

VITA

First Lieutenant Kevin G. Brown was born 4 January 1957 in Montezuma, Iowa. He graduated from Montezuma Community High School in 1975. He then attended Iowa State University in Ames, Iowa on a ROTC scholarship until 1979 receiving a Bachelor of Science Degree in Aeorspace Engineering. Upon graduation he received a commission in the USAF. While attending Iowa State he worked for The Boeing Commercial Airplane Company in Seattle, Wa. during the summer of 1978 In 1979, he worked in the preliminary design group at General Dynamics Corporation, Convair Division in San Diego, Ca. He entered active duty February 1980, serving with the 4950 th Test Wing, Wright-Patterson AFB, Oh until entering the School of Engineering, Air Force Institute of Technology in June 1982.

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release; Distribution Unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GAE/AA/ 83D-3	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION School of Engineering 6b. OFFICE SYMBOL (If applicable) AFIT/EN	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State and ZIP Code) Air Force Institute of Technology Wright-Patterson AFB, IH 45433	7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT NO.
11. TITLE (Include Security Classification) See Box 19	
12. PERSONAL AUTHOR(S) Kevin G. Brown, B.S., 1 Lt, USAF	
13a. TYPE OF REPORT MS Thesis 13b. TIME COVERED FROM ____ TO ____	14. DATE OF REPORT (Yr., Mo., Day) 83 December 15. PAGE COUNT
16. SUPPLEMENTARY NOTATION Lynn E. Wolaver Dean for Research and Professional Development Air Force Institute of Technology (AFIT) Wright-Patterson AFB, OH 45433	
17. COSATI CODES FIELD: 20 / GROUP: 04 / SUB. GR.	18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) Fluid Dynamics; Aerodynamic Characteristics; Nomerical Analysis; Adaptive Grid Beneration

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Title: Adaptive Grid Generation For Numerical Solution Of Partial Differential Equations

Thesis Advisor: James K. Hodge, Capt., USAF

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION UCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL James K. Hodge, Captain, USAF	22b. TELEPHONE NUMBER (Include Area Code) 513-255-3517 22c. OFFICE SYMBOL AFIT/ENY

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

SECURITY CLASSIFICATION OF THIS PAGE