

GRID EVOLUTION IN TIME ASYMPTOTIC PROBLEMS

Man Mohan Rai* and D. A. Anderson**
Iowa State University
Ames, Iowa

INTRODUCTION

Coordinate system selection is an important consideration in the time asymptotic numerical solution of any fluid flow or heat transfer problem. In solving such transient problems, the physical domain is usually transformed into a rectangular region with boundaries coincident with the physical boundaries. Once this transformation is completed, the transformed equations of motion are integrated until steady state is attained.

Most methods of generating systems of coordinates used in numerical solutions have been developed for elliptic problems. In these methods, the physical domain boundaries are known and the coordinate mesh is determined initially. Generally, the geometry of the mesh is not changed during the computation. Probably the most well known of these methods is the one developed by Thompson et al. (1) in which the transformed coordinates are obtained as a solution of Laplace's equation in physical space. A number of other investigators (2, 3, 4) have developed schemes which can be used to generate appropriate coordinate systems using the same general idea.

* Graduate Student

** Professor, Department of Aerospace Engineering

Unfortunately, the solution of a separate elliptic equation is not conveniently included in the solution of a time-dependent set of equations. Hindman et al. (5) solved the two-dimensional time-dependent Euler equations with a truly adaptive grid scheme. The grid motion in time was generated by taking the time derivative of the governing differential equations of the coordinate mapping which was the same as that developed by Thompson. This provided the necessary grid speed equations which were then integrated to obtain the grid motion as a function of time. Hindman's work did not consider techniques which might be used to modify the location of the interior points depending upon the local solution. The interior point motion depended solely upon boundary motion.

A technique for locating mesh points according to local flow information was presented by Dwyer et al. (6). This technique is similar to that used by Olson (7) and involves redistributing the mesh points at the end of any number of integration steps. This method does not permit a simple time integration of a differential equation similar to the equations of gas dynamics for the motion of the mesh points. It is the purpose of this paper to introduce a new technique which provides a simple way of moving the mesh points in physical space and reduces the error in the solution relative to that obtained using a fixed mesh.

Pierson et al. (8) have also worked on the generation of grids which minimize error, but their technique involves the solution of a minimization problem. The extension of such a method to higher dimensions with the accompanying increase in the number of mesh points

is not feasible due to the large amounts of computer time necessary to solve minimization problems. The method to be discussed in this paper is very simple in application and takes only a fraction of the time necessary to solve a minimization problem.

THE METHOD

To describe the basic idea employed in this paper, we consider transient problems in one space dimension. Let the physical space coordinates be x and t and let the computational space coordinates be ξ and τ where

$$\tau = t$$

$$\xi = \xi(x, t)$$

We require the calculation of the absolute value of the derivative ($|u_\xi|$) of some representative physical quantity (u) such as velocity, pressure, or temperature and the average value of the same derivative ($|u_\xi|_{av}$) for all mesh points. Given a certain number of grid points, truncation error can be minimized by allocating a number of points to the regions of large gradients and fewer points to the regions of small gradients. For an equispaced grid, a relocation of points in order to minimize error can be carried out. This can be achieved if points at which $|u_\xi|$ is larger than $|u_\xi|_{av}$ attract other points and points at which $|u_\xi|$ is smaller than $|u_\xi|_{av}$ repel other points. In other words, every point induces a velocity at every other point, the magnitude and direction depending upon the local 'excess gradient'. It is logical to assume that the further a point A is from a point B, the smaller the effect of point A on B.

This suggests that a $1/r^n$ law should be used. From the above considerations, it is possible to write

$$(\xi_i)_\tau = K \left[\sum_{j=i+1}^N \frac{[|u_\xi|_j - |u_\xi|_{av}]}{r_{i,j}^n} - \sum_{j=1}^{i-1} \frac{[|u_\xi|_j - |u_\xi|_{av}]}{r_{i,j}^n} \right]$$

$i = 2, 3 \dots (N-1)$ (1)

$$(x_\tau)_i = (\xi_i)_\tau / (\xi_x)_i$$
 (2)

where i is the point at which the velocity is being determined, 'N' is the total number of grid points, $r_{i,j}$ is the distance between points i and j in (ξ, τ) space and 'K' and 'n' are constants. The value of K can be determined if the maximum velocity that any point can achieve is specified. Convergence of the grid to a steady-state configuration is obtained by specifying a maximum value for K (K_{max}).

Strong analogies can be found between the present formulation and treating the grid points as point electrical charges whose individual charges are proportional to the local 'excess gradient'. The charges move so as to minimize the quantity

$$E = \sum_{j=1}^N [|u_\xi|_j - |u_\xi|_{av}]^2$$

the minimum value of E being zero.

The collapsing of two computational space points into one physical space point is not possible because of two reasons:

(a) The driving force g ,

$$g = |u_\xi| - |u_\xi|_{av} \quad (3)$$

becomes negative when two points get very close and, hence, the points begin to repel each other.

(b) The term ξ_x in Equation (2) gets very large as two points get very close. Hence, for a finite $(\xi_i)_T$, $(x_T)_i$ tends to zero; i.e., the closer two points get to each other, the more difficult it becomes for them to move toward each other. However, Equation (2) does not prevent extreme stretching of the mesh in physical space, thus giving rise to errors in the calculation of the transformation metrics. The details of preventing extreme stretching for the problems solved in this paper are presented in the section on results.

In the above discussion the driving force g is defined in terms of local and average first derivatives. A better formulation would be one in which g is defined in terms of quantities which are more representative of truncation error. One such quantity is the third derivative of u instead of the first derivative. The appropriate choice depends upon the order of the method being used and the problem itself. The flexibility in choosing the driving force and the quantity to be minimized is a particularly attractive feature of the current scheme.

Two constants K and n appear in Equation (1) and a third one, K_{\max} defines the maximum value that K can assume. The constants

K and K_{\max} together determine the grid speed. When K is less than K_{\max} , the grid speed is determined by K alone and when K is greater than K_{\max} , the grid speed is determined only by K_{\max} . At present these constants are chosen empirically. In choosing these constants one should bear in mind that very large values of K_{\max} result in grid oscillations which in turn result in longer convergence times, and very small values of K_{\max} result in low grid speeds and hence, once again longer convergence times are observed. The constant K is calculated by knowing the maximum velocity that any point can achieve in the computational space $\left[(\xi_i)_\tau \right]_{\max}$. The rules that govern the choice of $\left[(\xi_i)_\tau \right]_{\max}$ are the same as those that govern the choice of K_{\max} .

A variation of the constant 'n' between 1 and 8 did not make any difference in the final grid in the one-dimensional case studied and a small difference in the two-dimensional case. The number of iterations for convergence increases slightly when larger values of n are used. However, larger values of n imply a smaller range of influence for any given point. Consider a value of n ,

$$n = \frac{2}{\log(2)}$$

When $r = 2$,

$$\frac{1}{r^n} = 10^{-2}$$

This implies that only points adjacent to a given point make a significant contribution to the velocity of that point. Hence, Equation (1)

becomes

$$(\xi_i)_\tau \approx K \left\{ |u_\xi|_{i+1} - |u_\xi|_{i-1} \right\} \quad (4)$$

The use of Equation (4) instead of Equation (1) greatly speeds up the grid generation process.

EXTENSION TO MULTIDIMENSIONAL PROBLEMS

The method can be extended to problems in two and three space dimensions without any difficulty. In particular, for a problem in two space dimensions, let the physical coordinates be given by (x,y,t) and the computational coordinates by (ξ,η,τ) where

$$\tau = t$$

$$\xi = \xi(x,y,t)$$

$$\eta = \eta(x,y,t)$$

We now require the calculation $|u_\xi|$ and $|u_\eta|$ for every point and $|u_\xi|_{av}$ for every row of points and $|u_\eta|_{av}$ for every column of points as in Figure 1. The grid speed equations are given by

$$(\xi_{i,j})_\tau = K_1 \sum_{\ell=1}^M \left[\sum_{k=i+1}^N \frac{[|u_\xi|_{k,\ell} - |u_\xi|_{av_\ell}]}{r^n} - \sum_{k=1}^{i-1} \frac{[|u_\xi|_{k,\ell} - |u_\xi|_{av_\ell}]}{r^n} \right]$$

$$\begin{aligned}
 (\eta_{i,j})_{\tau} = & K_2 \sum_{k=1}^N \left[\sum_{\ell=j+1}^M \frac{[|u_{\eta}|_{k,\ell} - |u_{\eta}|_{av_k}]}{r^n} \right. \\
 & \left. - \sum_{\ell=1}^{j-1} \frac{[|u_{\eta}|_{k,\ell} - |u_{\eta}|_{av_k}]}{r^n} \right]
 \end{aligned} \tag{5}$$

$$r = \sqrt{(i-k)^2 + (j-\ell)^2}$$

where K_1, K_2 and n are constants, N the number of points in the ξ direction and M the number of points in the η direction. The values of K_1 and K_2 can be determined by specifying $\left[(\xi_{i,j})_{\tau} \right]_{\max}$ and $\left[(\eta_{i,j})_{\tau} \right]_{\max}$ respectively. Grid convergence can be achieved by specifying $(K_1)_{\max}$ and $(K_2)_{\max}$ as in the one-dimensional case.

We also have the relationships

$$(\xi_{i,j})_{\tau} = (\xi_{x\tau} + \xi_{y\tau})_{i,j} \tag{6}$$

$$(\eta_{i,j})_{\tau} = (\eta_{x\tau} + \eta_{y\tau})_{i,j}$$

which yield

$$(x_{\tau})_{i,j} = \frac{[(\eta_y)_{i,j}(\xi_{i,j})_{\tau} - (\xi_y)_{i,j}(\eta_{i,j})_{\tau}]}{J}$$

$$(y_\tau)_{i,j} = \frac{[(\xi_x)_{i,j}(\eta_{i,j})_\tau - (\eta_x)_{i,j}(\xi_{i,j})_\tau]}{J} \quad (7)$$

$$J = \xi_x \eta_y - \eta_x \xi_y$$

From Equation (7) it can be seen that the collapsing of mesh points and the overlapping of grid lines is again prevented as in the one-dimensional case.

Points lying along a constant η line can be made to move tangential to this line by specifying $(\eta_{i,j})_\tau$ to be zero for all these points. A similar procedure can be adopted for constant ξ lines. This facilitates the movement of points along surface boundaries, etc. However, this type of unnatural constraint on the velocity of points leads to a slightly distorted grid as shown in Figure 2. A more natural way of making points move tangential to boundaries is to specify periodic boundaries and use the pseudo points outside the region of interest also to calculate the grid speed. This procedure of calculating the grid speed results in the grid shown in Figure 3. The distortions present in Figure 2 are absent in Figure 3 and the grid is seen to be smooth and uniform. The grids shown in Figures 2 and 3 were generated using a known solution to the two-dimensional transient, linear, viscous Burger's equation.

RESULTS

The first problem solved using the present grid generation technique was the one-dimensional unsteady viscous Burger's equation

$$u_t + uu_x = \mu u_{xx} \quad (8)$$

with the initial condition

$$u(0,x) = \begin{cases} 1 & x = 0 \\ 0 & 0 < x \leq 1 \end{cases} \quad (9)$$

and the boundary conditions

$$\begin{aligned} u(t,0) &= 1 \\ u(t,1) &= 0 \end{aligned} \quad (10)$$

This problem has the steady state solution

$$u = \bar{u} \tanh \left[\frac{\text{Re}}{2} (1-x) \right] \quad (11)$$

where

$$\text{Re} = 1/\mu \quad \begin{array}{l} \text{ORIGINAL PAGE IS} \\ \text{OF POOR QUALITY} \end{array} \quad (12)$$

and \bar{u} is the solution of the equation

$$\frac{\bar{u}-1}{\bar{u}+1} = \exp \{-\bar{u}\text{Re}\} \quad (13)$$

The slope of the steady state solution at the right end increases and that at the left end tends to zero as Re increases.

McCormack's method was used to integrate Equation (8) and three point central differences were used to calculate the metrics of the transformation. The stability limit for McCormack's method for this problem was determined using the empirical formula given by Tannehill et al. (9).

Results are presented for various values of Re in Figures 4-8. In all cases the steady state results using an adaptive grid and those obtained using an equispaced grid are compared with the exact solution. In Figure 4 results for $Re = 1$ are shown. The errors are very small ($< 0.04\%$) in both cases but the peak error without an adaptive grid is about 1.82 times the peak error with an adaptive grid. In Figure 5 results are presented for $Re = 2$. The ratio of the peak errors is now about 4.90 and a significant improvement in accuracy is seen. However, in Figure 5, the adaptive grid shows a slightly larger error in the region $0 < x < 0.2$. This is due to the fact that the second point in the grid has moved to the right a substantial distance resulting in a higher error in this region.

Figure 6 presents results for $Re = 3$. The second point in this case moves so far to the right that the truncation error in calculating the transformation metrics in this region swamps the entire solution resulting in a solution that is worse than the one obtained using an equispaced grid. In order to prevent extreme stretching of the grid it is necessary to include a measure of the truncation error introduced in calculating the transformation metrics into the driving force g ,

$$g = |x_\xi| - |x_\xi|_{av} + \theta \{ |u_\xi| - |u_\xi|_{av} \} \quad (14)$$

where θ is a constant. Since x_ξ is greater than zero and u_ξ is less than zero in this problem, Equation (14) can be written as

$$g = x_\xi - (x_\xi)_{av} - \theta \{ u_\xi - (u_\xi)_{av} \} \quad (15)$$

Since the grid converges when g is a constant over the entire region, the transformation for the converged grid can be shown to be

$$\xi = 1 - fu - (1-f)(1-x) \quad 0 \leq f \leq 1 \quad (16)$$

where f is a constant. Hence, an equivalent way of preventing extreme stretching is to define \bar{u} as

$$\bar{u} = fu + (1-f)(1-x) \quad (17)$$

and the driving force g as

$$g = |\bar{u}_\xi| - |\bar{u}_\xi|_{av} \quad (18)$$

The error curve obtained for $Re = 3$ and $f = 0.7$ is also shown in Figure 6. A substantial decrease in error is seen, the ratio of the peak errors being about 3.80. Figures 7 and 8 present results for $Re = 5$ and $Re = 10$ respectively. In both cases a smoothed form of the solution as given by Equation (17) is used. The ratio of peak errors is about 2.23 for $Re = 5$ and 2.13 for $Re = 10$. Figure 9 shows the transformation obtained for the case $Re = 3$, $f = 0.7$. The uniform nature of the transformation is apparent.

A better measure of the total truncation error at a point (e) is

$$e \propto dx^2 u_{xxx} \quad (19)$$

which can be approximated in this case as

$$e \propto dx^2 u_x \quad (20)$$

which yields

$$e \propto u_{\xi}/\xi_x \quad (21)$$

Equation (21) suggests a driving force of the form

$$g = |u_{\xi}/\xi_x| - |u_{\xi}/\xi_x|_{av} \quad (22)$$

Results of using such a driving force for the case $Re = 3$ are presented in Figure 10. The errors obtained are comparable to the ones obtained using an optimal f . However, the advantage in using this new form of the driving force lies in eliminating the empiricism required in determining the optimal f . Similar results were obtained for all $Re < 5.0$. Excessive stretching was once again observed for higher values of Re , indicating the inaccuracy in estimating the error. The analysis and results presented in this and the preceding paragraph show that the method is limited only by the accuracy with which the total truncation error at a point can be estimated.

The second problem solved was the two-dimensional unsteady, linearized, viscous Burger's equation

$$u_t + u_x + u_y = \mu(u_{xx} + u_{yy}) \quad (23)$$

in a square domain with the initial conditions

$$u(x,0,0) = 1 + \frac{1 - \exp(\text{Re}(x-1))}{(1 - \exp(-\text{Re}))} \quad (24)$$

$$u(0,y,0) = 1 + \frac{1 - \exp(\text{Re}(y-1))}{(1 - \exp(-\text{Re}))}$$

$$u = 1 \quad \text{otherwise}$$

where

$$\text{Re} = 1/\mu \quad (25)$$

and the boundary conditions

$$u(x,0,t) = 1 + \frac{1 - \exp(\text{Re}(x-1))}{(1 - \exp(-\text{Re}))}$$

$$u(0,y,t) = 1 + \frac{1 - \exp(\text{Re}(y-1))}{(1 - \exp(-\text{Re}))} \quad (26)$$

$$u(x,1,t) = 1$$

$$u(1,y,t) = 1$$

This problem has the steady state solution

$$u = 1 + \frac{(1 - \exp(\text{Re}(x-1))) (1 - \exp(\text{Re}(y-1)))}{(1 - \exp(-\text{Re}))^2} \quad (27)$$

McCormack's method was used to integrate Equation (23) and three point central differences were used to calculate the metrics of the transformation. To prevent excessive stretching of the grid a smoothed version of the solution (\bar{u})

$$\bar{u} = fu + (1-f) (4-x-y)/2 \quad 0 \leq f \leq 1 \quad (28)$$

is used to calculate the driving force.

Figure 11 shows the grid obtained for $Re = 5$ and $f = 0.3$. The error is calculated at the points shown in Figure 11 and a linear interpolation is used to calculate the error at the points corresponding to the equispaced grid. The results are presented in Figures 12-15, at each y station. The adaptive grid yields slightly higher errors in the low gradient region as in Figure 12 and gradually progresses to much lower errors in the high gradient regions as in Figure 15. The increases in accuracy are not as high as in the one-dimensional case, the main reason being the inaccuracy in establishing the local truncation error. One complication that exists only in two- and three-dimensional problems is the appearance of cross derivative terms in any estimate of the local truncation error. The absence of cross derivative terms in the present formulation of the grid generation scheme is felt particularly at the point $x = 0.8, y = 0.2$ in Figure 12. This point has a large value of u_x and a small value of u_y resulting in mesh clustering only in the x direction. However, the terms u_{xyy} and u_{xxy} are by no means small and hence due to large Δy in this region give rise to large errors. Future work with two-dimensional problems will require that the influence of cross derivative terms be included in the generation of grids.

TIME REQUIREMENTS

The number of integration steps required for convergence is always greater with an adaptive grid because of the lower values of maximum allowable time steps associated with mesh clustering. The ratio of the number of steps required with and without an adaptive grid goes all the way from 3.4 for $Re = 10$ to 1.4 for $Re = 1$ in the one-dimensional case and takes on a value of 2.3 in the two-dimensional case. However, time estimates will be given only on a per integration step basis. In the one-dimensional case the generation of the grid and recalculation of the transformation metrics takes less than 10% of the time taken for integration. In the two-dimensional case, the generation of the grid takes 25% and recalculation of metrics takes 70% of the time taken for integration. One of the reasons for the excessive time taken for the calculation of metrics is the presence of second derivatives like ξ_{xx} , ξ_{yy} , η_{xx} , and η_{yy} , all of which need to be determined numerically. The absence of these second derivatives greatly speeds up the calculation of metrics. Furthermore, if the problem requires the recalculation of metrics even without an adaptive grid, as in shock fitting programs, the time required to use an adaptive grid becomes very attractive. It must also be remembered that the percent extra time in this case is high because the equation being solved is very simple. Since the time for grid generation remains about the same in far more complicated problems, the present extra time for grid generation will be much less for such problems.

In conclusion, the major contributions of this paper are:

(a) Formulation of simple first order partial differential equations for the grid point velocity in transient problems.

(b) Significant error reductions for solutions of Burger's equation in one and two dimensions.

(c) The use of local flow information and boundary motion in determining the interior grid point motion.

REFERENCES

1. Thompson, J. F.; Thames, F. C.; and Mastin, C. W.: Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate Systems for Field Containing Any Number of Arbitrary Two-Dimensional Bodies. *J. Comp. Phys.*, 15, 299, 1974.
2. Thompson, J. F.; Thames, F. C.; and Mastin, C. W.: TOMCAT - A Code for Numerical Generation of Boundary-Fitted Curvilinear Coordinate Systems on Fields Containing Any Number of Arbitrary Two-Dimensional Bodies. *J. Comp. Phys.*, vol. 24, no. 3, 1977.
3. Thames, F. C.: Numerical Solution of the Incompressible Navier-Stokes Equations about Arbitrary Two-Dimensional Bodies. Ph.D. Dissertation, Mississippi State Univ., Mississippi State, MS, 1975.
4. Middlecoff, J. F.; and Thomas, P. D.: Direct Control of the Grid Point Distribution in Meshes Generated by Elliptic Equations. AIAA Paper 79-1462, July 1979.
5. Hindman, R. G.; Kutler, P.; and Anderson, D. A.: A Two-Dimensional Unsteady Euler-Equation Solver for Flow Regions With Arbitrary Boundaries. AIAA Paper 79-1465, July 1979.
6. Dwyer, H. A.; Kee, R. J.; and Sanders, B. R.: An Adaptive Grid Method for Problems in Fluid Mechanics and Heat Transfer. AIAA Paper 79-1464, July 1979.
7. Olsen, J.: Subsonic and Transonic Flow Over Sharp and Round Nosed Nonlifting Airfoils. Ph.D. Dissertation, Ohio State University, Columbus, Ohio, 1976.
8. Pierson, B. L.; and Kutler, P.: Optimal Nodal Point Distribution for Improved Accuracy in Computational Fluid Dynamics. AIAA Paper 79-0272, January 1979.
9. Tannehill, J. C.; Holst, T. L.; and Rakich, J. V.: Numerical Solution of Two-Dimensional Viscous Blunt Body Flows with an Impinging Shock. AIAA Paper 75-154, January 1975.

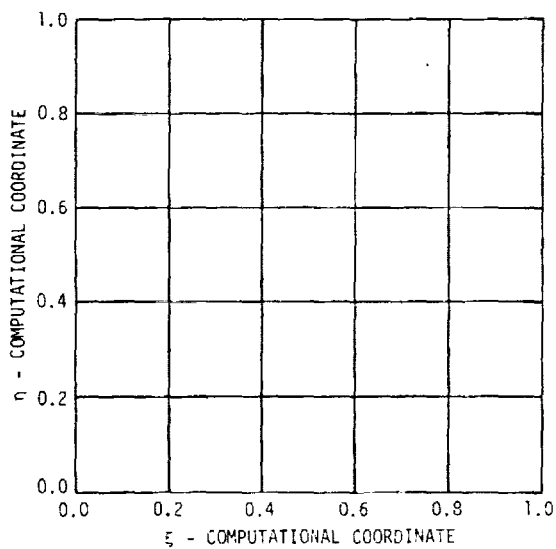


Fig. 1. Computational space.

ORIGINAL PAGE IS
OF POOR QUALITY

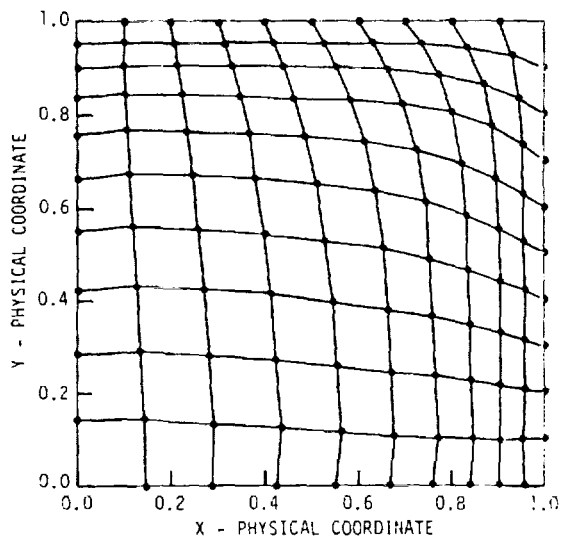


Fig. 2. Grid generated using aperiodic boundaries.

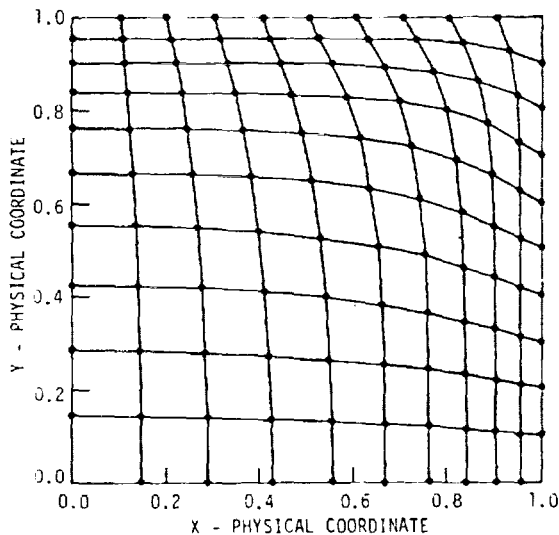


Fig. 3. Grid generated using periodic boundaries.

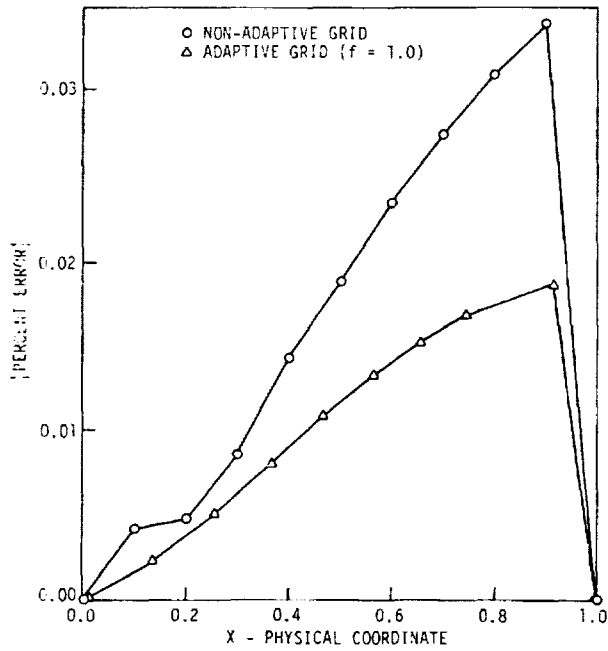


Fig. 4. Comparison of errors,
Re = 1.0.

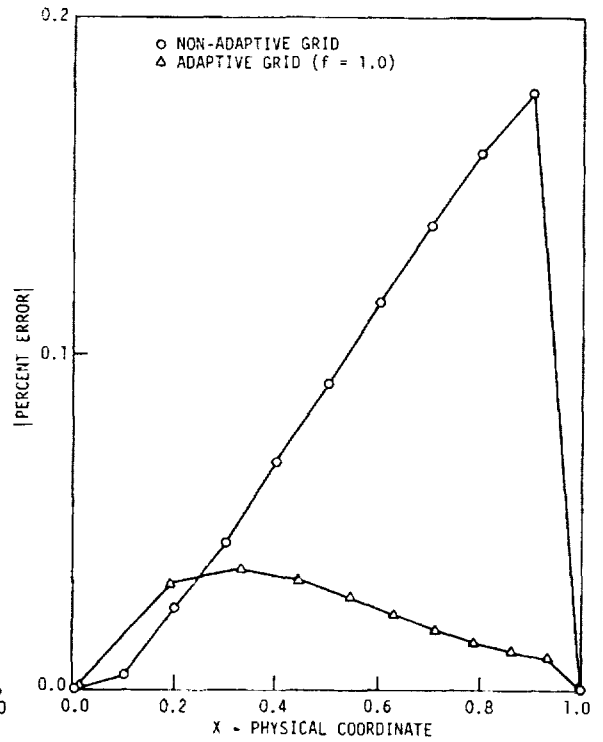


Fig. 5. Comparison of errors,
Re = 2.0.

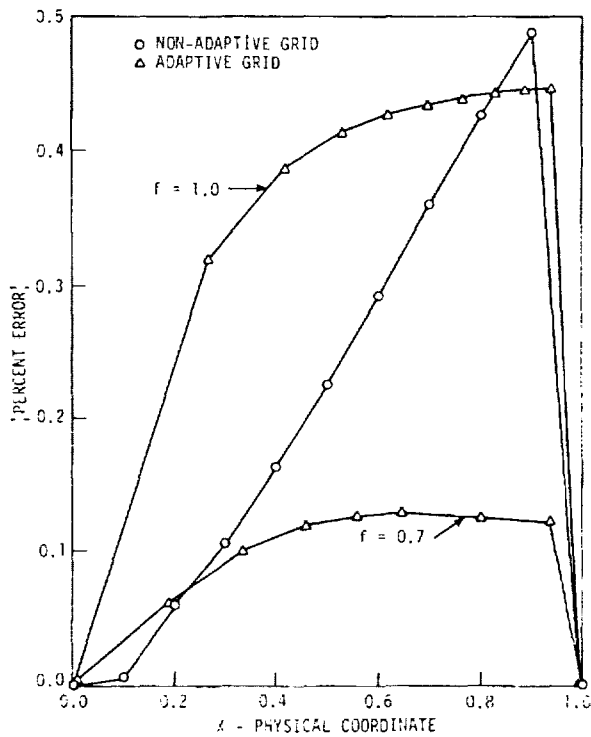


Fig. 6. Comparison of errors,
Re = 3.0.

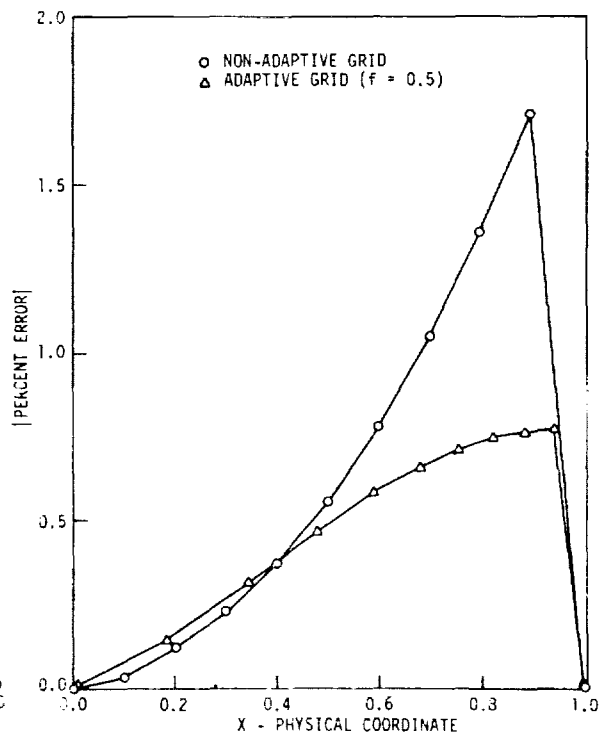


Fig. 7. Comparison of errors,
Re = 5.0.

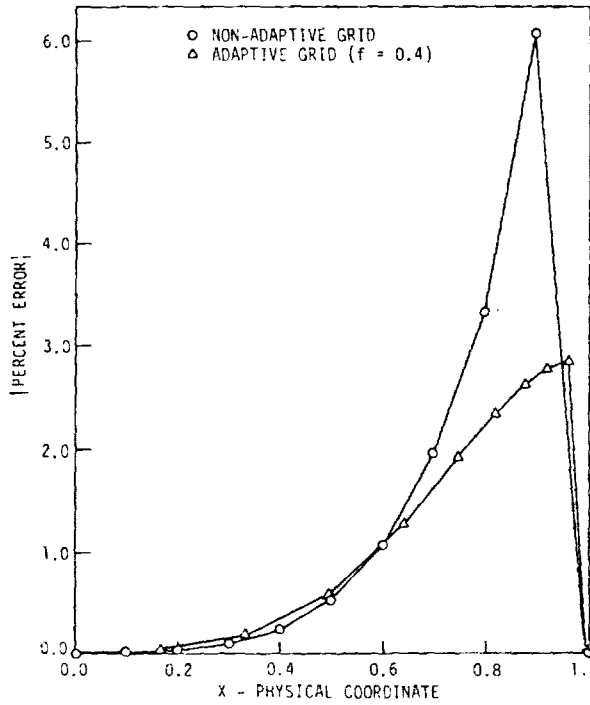


Fig. 8. Comparison of errors, $Re = 10.0$.

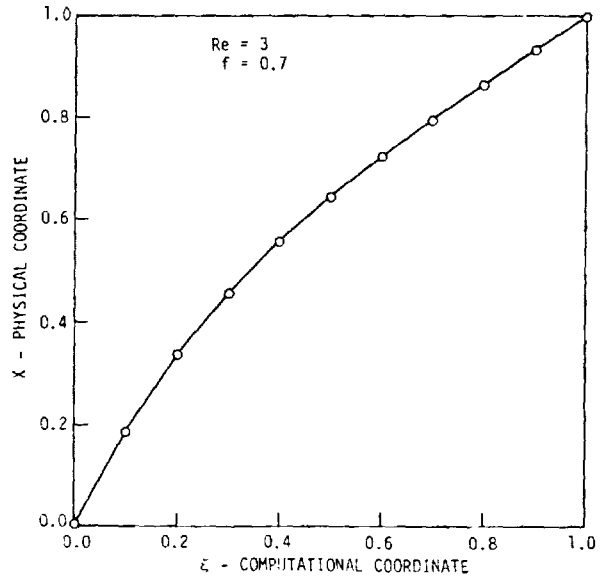


Fig. 9. Converged grid for the one-dimensional Burger's equation.

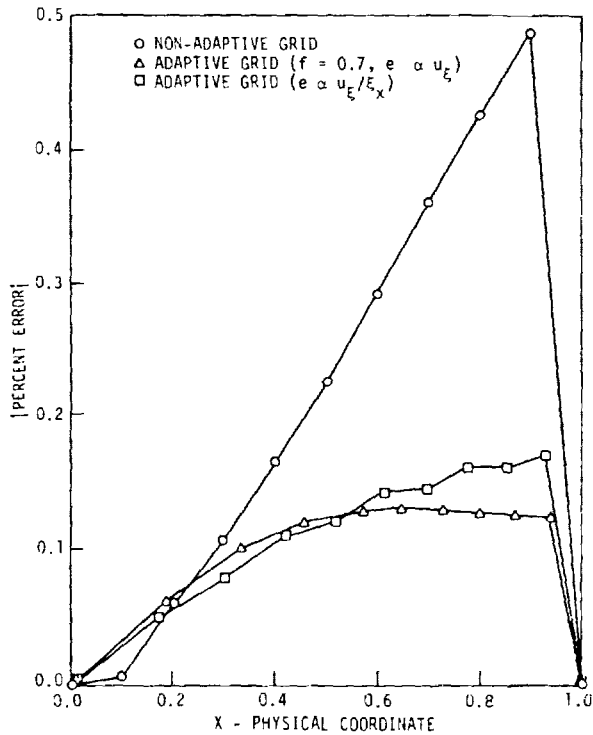


Fig. 10. Comparison of errors, $Re = 3.0$.

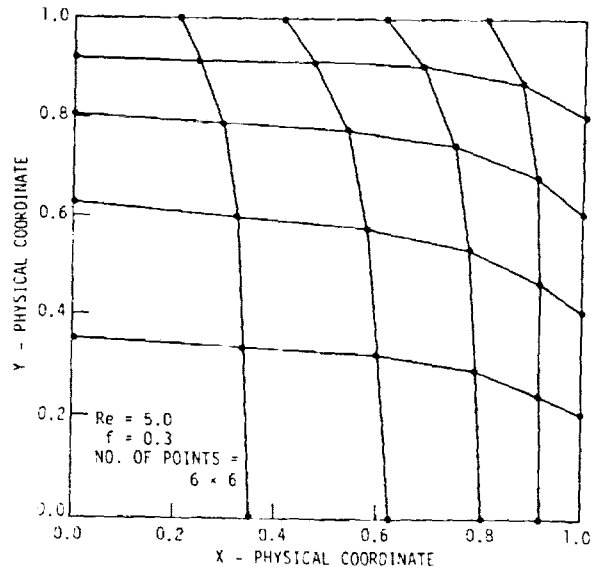


Fig. 11. Converged grid for the two-dimensional Burger's equation.

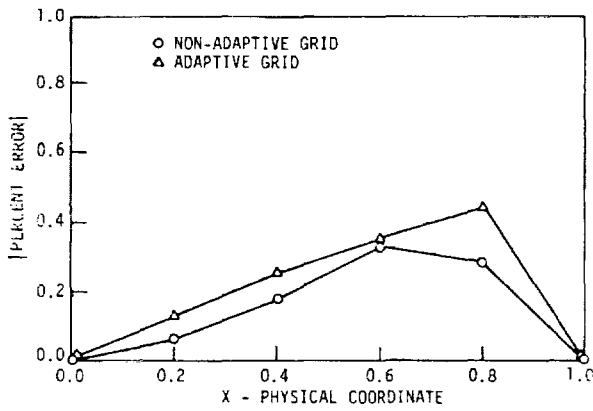


Fig. 12. Comparison of errors, $Re = 5.0, Y = 0.2.$

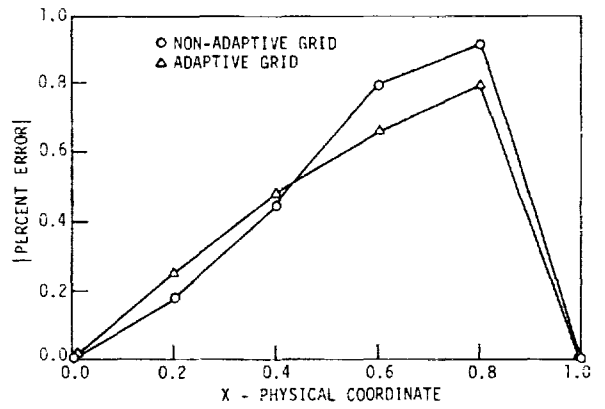


Fig. 13. Comparison of errors, $Re = 5.0, Y = 0.4.$

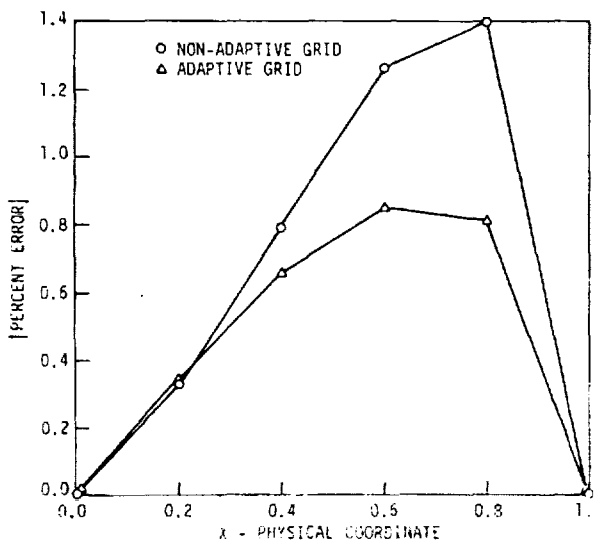


Fig. 14. Comparison of errors, $Re = 5.0, Y = 0.6.$

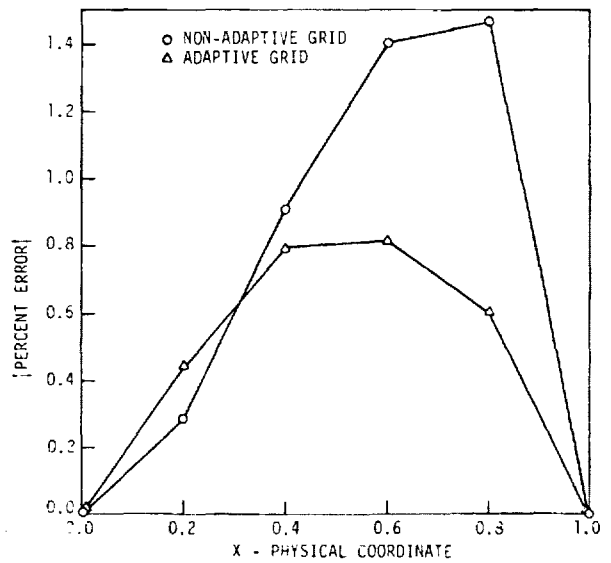


Fig. 15. Comparison of errors, $Re = 5.0, Y = 0.8.$