JOURNAL OF COMPUTATIONAL PHYSICS 43, 327–344 (1981)

Grid Evolution in Time Asymptotic Problems

MAN MOHAN RAI AND D. A. ANDERSON

Department of Aerospace Engineering and Computational Fluid Dynamics Institute, Iowa State University, Ames, Iowa 50011

Received November 12, 1980; revised June 5, 1981

The selection of the proper coordinate system in solving any fluid flow or heat transfer problem is a very important consideration. A new technique of moving mesh points in physical space is introduced so as to reduce the error in a computed asymptotic solution relative to that obtained using a fixed mesh. The technique has been used to solve the simple viscous Burgers' equation in one and two dimensions. Substantial error reductions are demonstrated. The treatment of boundary points and the effect of using different error criteria in generating grids are discussed.

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 involve the solution of systems of elliptic partial differential equations. 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 [2–4] have developed schemes which can be used to generate appropriate coordinate systems using the same general idea.

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

0021-9991/81/100327-18$02.00/0

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 Olsen [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 and Kutler [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, 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, we assume that truncation error can be reduced by allocating a number of points to the regions of large gradients and fewer points to the regions of small gradients. A relocation of points in order to reduce error can be carried out very easily using the equispaced grid in the computational domain. 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

$$(\xi_i)_\tau = K\left[\sum_{j=i+1}^{N} \frac{[|u_\xi|_j - |u_\xi|_{av}]}{r_{i,j}^n} - \sum_{j=1}^{i-1} \frac{[|u_\xi|_j - |u_\xi|_{av}]}{r_{i,j}^n}\right], \quad i = 2, 3, \ldots, (N-1), \tag{1}$$

$$(x_\tau)_i = (\xi_i)_\tau / (\xi_x)_i, \tag{2}$$

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.

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 for two reasons:

(a) The driving force g,

$$g = |u_\xi| - |u_\xi|_{av} \tag{3}$$

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

(b) The term ξ_x in Eq. (2) gets very large as two points get very close. Hence, for a finite $(\xi_i)_\tau$, $(x_\tau)_i$ 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.

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. 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.

The quantities $(\xi_i)_\tau$ and $[|u_\xi| - |u_\xi|_{av}]/r^n$ in Eq. (1) are physically unrelated and hence the scaling factor K is necessary to equate them. The factor K is chosen such that the velocity at any point does not exceed a preset maximum $[(\xi_i)_\tau]_{max}$. Hence, in order to calculate K, we calculate the grid speed at every point assuming $K = 1$ and then calculate K as

$$K = [(\xi_i)_\tau]_{max} / |(\xi_i)_\tau|_{\text{calculated max}}.$$

The grid velocity at every point is now rescaled as

$$[(\xi_i)_\tau]_{\text{rescaled}} = K[(\xi_i)_\tau]_{\text{calculated}}.$$

As the calculation proceeds, $[|u_\xi| - |u_\xi|_{\text{av}}]$ at every point becomes smaller and hence K becomes larger in order that $|(\xi_i)_\tau|$ at some point in the grid achieves the value $[(\xi_i)_\tau]_{\max}$. The restriction of K to some preset maximum $K_{\max}$ results in grid velocities that die out quickly. Hence the convergence of the grid to a steady-state configuration can be obtained by specifying a maximum value for $K(K_{\max})$. A mathematical proof of convergence has not been attempted for this type of convergence because in most practical problems the grid point velocities are exponentially damped out for reasons given later in this section.

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 $[(\xi_i)_\tau]_{\max}$. The rules that govern the choice of $[(\xi_i)_\tau]_{\max}$ are the same as those that govern the choice of $K_{\max}$.

The determination of local truncation error involves the calculation of higher-order derivatives. For example, when a second-order method is used to calculate the solution to a first-order differential equation, a measure of the local truncation error inluces includes the third derivative of the dependent variable (u). Since the solution is $O(\Delta x^2)$ accurate, the calculation of u_{xxx}, which involves a division by Δx^3, is questionable. Hence we use an approximation to the error which includes only lower-order derivatives. This approximation often results in excessive stretching in some regions. This phenomenon can be prevented by prescribing the minimum and maximum values for the Jacobian of the transformation (J) at every point. Grid point velocities can be damped out exponentially as the limits of J are approached. Specifying limits for J introduces further empiricism into the problem. However, this cannot be avoided until better methods of evaluating local truncation error are developed.

A variation of the constant n in Eq. (1) between 1 and 8 did not make any difference in the final grid in the one-dimensional case studied but did make 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,

$$n = 2/\log(2).$$

When $r = 2$,

$$1/r^n = 10^{-2}.$$

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

$$(\xi_i)_\tau \simeq K\{|u_\xi|_{i+1} - |u_\xi|_{i-1}\}. \tag{4}$$

The use of Eq. (4) instead of Eq. (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 $|u_\xi|_{\text{av}}$ for every row of points and $|u_\eta|_{\text{av}}$ for every column of points as in Fig. 1. The grid speed equations are given by

$$\begin{aligned}
(\xi_{i,j})_\tau &= K_1 \sum_{l=1}^{M} \left[\sum_{k=i+1}^{N} \frac{[|u_\xi|_{k,l} - |u_\xi|_{\text{av}_l}]}{r^n} - \sum_{k=1}^{i-1} \frac{[|u_\xi|_{k,l} - |u_\xi|_{\text{av}_l}]}{r^n} \right], \\
(\eta_{i,j})_\tau &= K_2 \sum_{k=1}^{N} \left[\sum_{l=j+1}^{M} \frac{[|u_\eta|_{k,l} - |u_\eta|_{\text{av}_k}]}{r^n} - \sum_{l=1}^{j-1} \frac{[|u_\eta|_{k,l} - |u_\eta|_{\text{av}_k}]}{r^n} \right], \\
r &= \sqrt{(i-k)^2 + (j-l)^2}.
\end{aligned} \tag{5}$$

where K_1, K_2 and n are constants, N is the number of points in the ξ direction and M

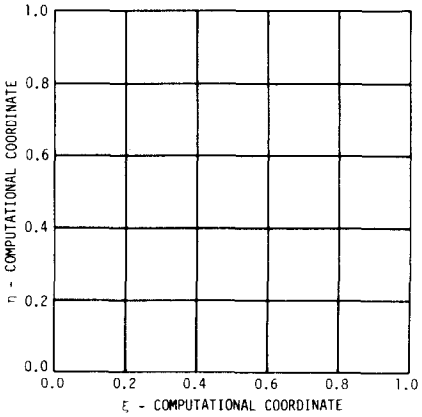

FIG. 1. Computational space.

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

We alse have the relationships

$$\begin{aligned}(\xi_{i,j})_\tau &= (\xi_x x_\tau + \xi_y y_\tau)_{i,j},\\ (\eta_{i,j})_\tau &= (\eta_x x_\tau + \eta_y y_\tau)_{i,j}\end{aligned} \tag{6}$$

which yield

$$\begin{aligned}(x_\tau)_{i,j} &= \frac{[(\eta_y)_{i,j}(\xi_{i,j})_\tau - (\xi_y)_{i,j}(\eta_{i,j})_\tau]}{J},\\ (y_\tau)_{i,j} &= \frac{[(\xi_x)_{i,j}(\eta_{i,j})_\tau - (\eta_x)_{i,j}(\xi_{i,j})_\tau]}{J},\\ J &= \xi_x\eta_y - \eta_x\xi_y.\end{aligned} \tag{7}$$

From Eq. (7) it can be seen that the collapsing of mesh points and the overlapping of grid lines are 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 Fig. 2. Consider a one-dimensional problem in which the quantity $[|u_\xi| - |u_\xi|_{\text{av}}]$ is almost constant in the region of the left boundary. If we were to calculate the velocity of the second grid point using Eq. (1) with $K = 1$ and $n = 1$,

$$(\xi_2)_\tau = [|u_\xi| - |u_\xi|_{\text{av}}]\left\{\frac{1}{r} + \frac{1}{2r} + \frac{1}{3r}\cdots - \frac{1}{r}\right\}$$

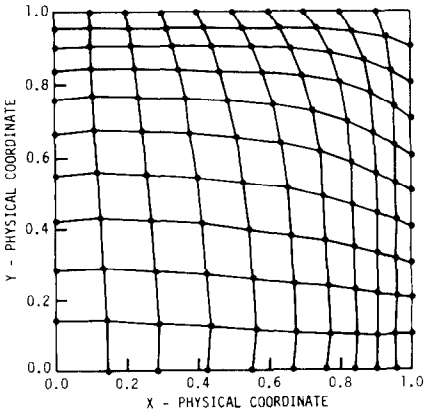

FIG. 2. Grid generated using aperiodic boundaries.

and we get

$$(\xi_2)_\tau = \frac{[|u_\xi| - |u_\xi|_{av}]}{r} \left\{ \frac{1}{2} + \frac{1}{3} \cdots \right\} > 0.$$

However, the value of $(\xi_2)_\tau$ must be zero since it is in a constant $[|u_\xi| - |u_\xi|_{av}]$ region. This problem can be overcome by assuming periodic boundary conditions, i.e., a set of pseudo points outside the left boundary such that

$$|u_\xi|_j = |u_\xi|_{2-j}, \qquad j = 0, -1, \ldots,$$

and using the pseudo points also in calculating grid speeds. This also results in the requirement that

$$(\xi_1)_\tau = 0.$$

The same considerations apply to the right boundary. A similar situation exists in two dimensions. A natural way of making points move tangential to boundaries is to specify periodic boundaries and use pseudo points outside the region of interest to calculate the grid speed. This procedure of calculating the grid speed results in the grid shown in Fig. 3. The distortions present in Fig. 2 are absent in Fig. 3 and the grid is seen to be smooth and uniform. The grids shown in Figs. 2 and 3 were generated using a known solution to the two-dimensional transient, linear, viscous Burgers' equation.

RESULTS

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

$$u_t + uu_x = \mu u_{xx} \tag{8}$$

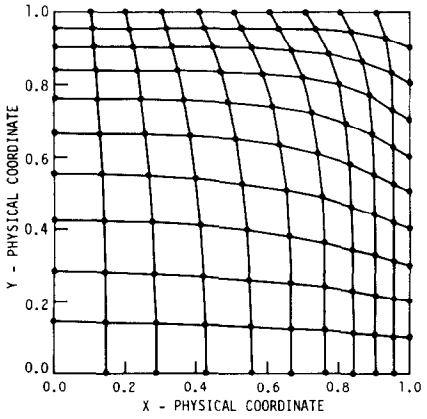

FIG. 3. Grid generated using periodic boundaries.

with the initial condition

$$\begin{aligned} u(0, x) &= 1, \qquad x = 0 \\ &= 0, \qquad 0 < x \leqslant 1, \end{aligned} \tag{9}$$

and the boundary conditions

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

This problem has the steady-state solution

$$u = \hat{u} \tanh\left[\frac{\hat{u}Re}{2}(1 - x)\right], \tag{11}$$

where

$$Re = 1/\mu \tag{12}$$

and $\hat{u}$ is the solution of the equation

$$\frac{\hat{u} - 1}{\hat{u} + 1} = \exp[-\hat{u}Re]. \tag{13}$$

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

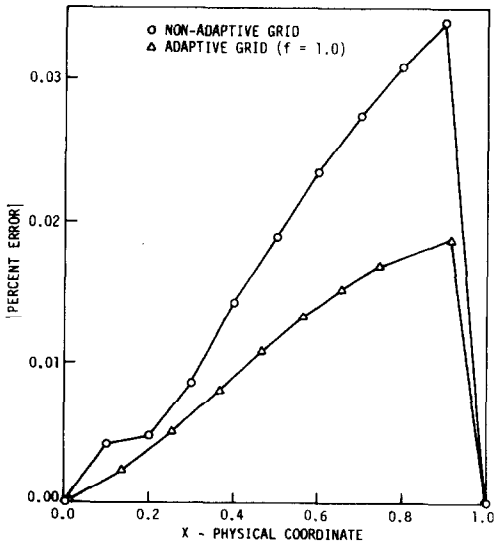

FIG. 4. Comparison of errors for the one-dimensional viscous Burgers' equation, $Re = 1.0$.

MacCormack's [9] method was used to integrate Eq. (8) and three point central differences were used to calculate the metrics of the transformation. The stability limit for MacCormack's method for this problem was determined using the empirical formula given by Tannehill *et al.* [10]. Equation (4) was used to determine grid point velocities.

Results are presented for various values of Re in Figs. 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 Fig. 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 Fig. 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 Fig. 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 inaccuracy in estimating local truncation error as equal to $|u_\xi|$ is apparent in this case. The errors in the terms u_ξ and $u_{\xi\xi}$ are better approximated as

$$e(u_\xi) \simeq -\tfrac{1}{2}x_\xi x_{\xi\xi} u_{xx} - \tfrac{1}{6}x_\xi^3 u_{xxx}, \tag{14}$$

$$e(u_{\xi\xi}) \simeq -\tfrac{1}{2}x_\xi^2 x_{\xi\xi} u_{xxx} - \tfrac{1}{12}x_\xi^4 u_{xxxx}. \tag{15}$$

A rapidly varying x_ξ in the grid results in large values of $x_{\xi\xi}$ and hence in larger errors in the computed solution. The rapid variation of x_ξ can be prevented as discussed earlier by limiting the change in Jacobian at every point. A natural way of preventing excessive stretching in this particular problem is to define $\bar{u}$ as

$$\bar{u} = fu + (1-f)(1-x), \qquad 0 \leqslant f \leqslant 1, \tag{16}$$

and the driving force g as

$$g = |\bar{u}_\xi| - |\bar{u}_\xi|_{\text{av}}. \tag{17}$$

Note that even though the definition of $\bar{u}$ is problem dependent, the idea of exponentially damping out grid velocities as the Jacobian of the transformation approaches a prescribed limit is problem independent. It just happens that there is an easier way of preventing excessive point motion in this case. The error curve obtained for $Re = 3$ and $f = 0.7$ is also shown in Fig. 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 Eq. (16) 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.

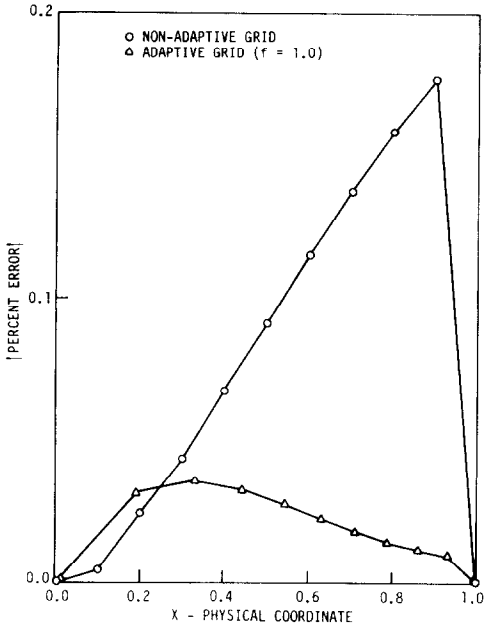

FIG. 5. Comparison of errors for the one-dimensional viscous Burgers' equation, $Re = 2.0$.

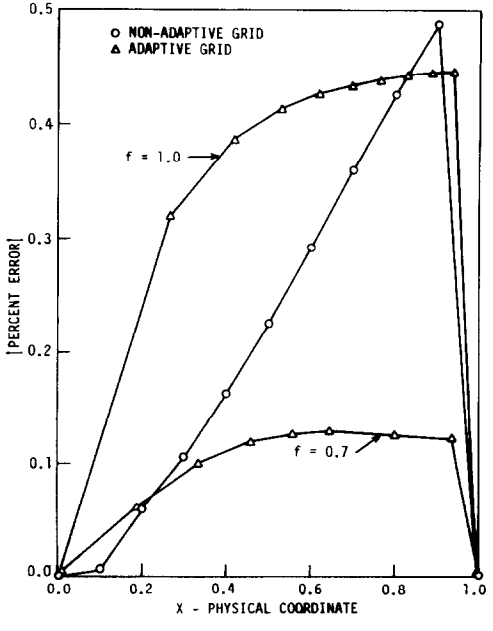

FIG. 6. Comparison of errors for the one-dimensional viscous Burgers' equation, $Re = 3.0$.

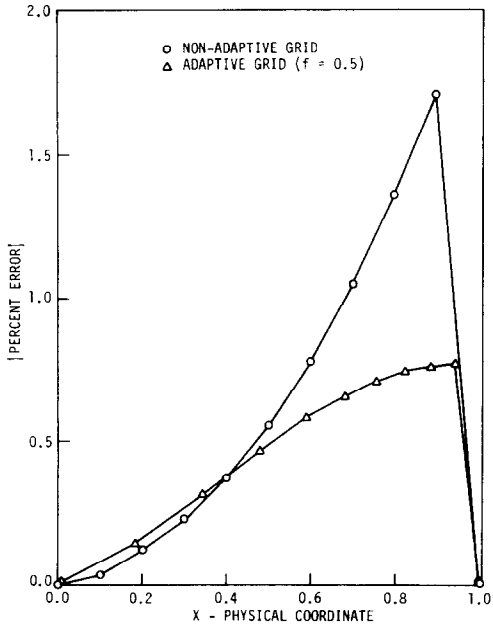

FIG. 7. Comparison of errors for the one-dimensional viscous Burgers' equation, $Re = 5.0$.

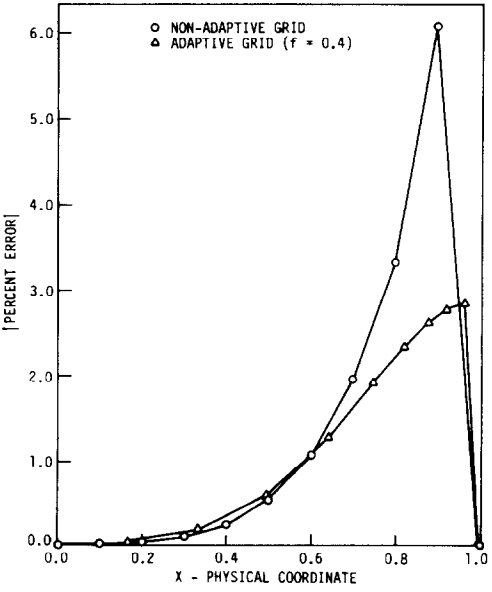

FIG. 8. Comparison of errors for the one-dimensional viscous Burgers' equation, $Re = 10.0$.

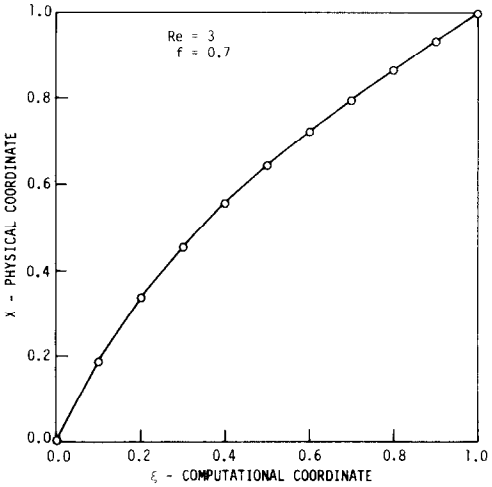

FIG. 9. Converged grid for the one-dimensional Burgers' equation.

An equivalent form of Eq. (14) is given by

$$e(u_x) \simeq -\tfrac{1}{2}x_{\xi\xi}u_{xx} - \tfrac{1}{6}x_\xi^2 u_{xxx}. \tag{18}$$

Assuming $x_{\xi\xi}$ to be small we get

$$e(u_x) \simeq -\tfrac{1}{6}x_\xi^2 u_{xxx}. \tag{19}$$

Since the evaluation of u_{xxx} using computed values of u is problematic, we make one further approximation

$$|u_x| \propto |u_{xxx}| \tag{20}$$

which yields

$$|e| \propto |u_\xi/\xi_x| \tag{21}$$

instead of the much simpler form $|e| \propto |u_\xi|$ which was used earlier. Equation (21) suggests a driving force of the form

$$g = |u_\xi/\xi_x| - |u_\xi/\xi_x|_{\text{av}}. \tag{22}$$

Results of using such a driving force for the case $Re = 3$ are presented in Fig. 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

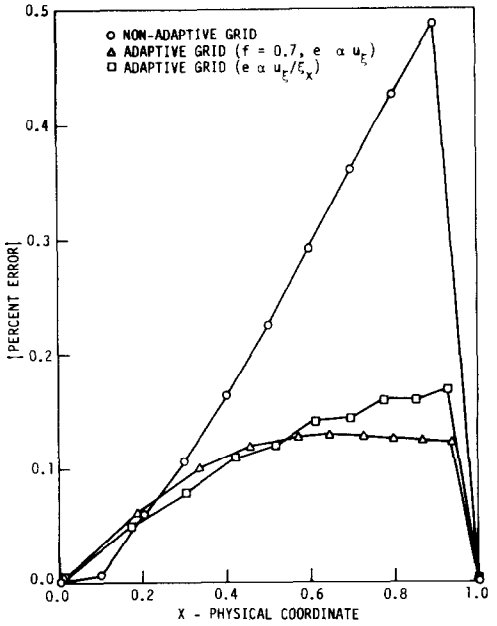

FIG. 10. Comparison of errors for the one-dimensional viscous Burgers' equation, $Re = 3.0$.

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 Burgers' equation

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

in a square domain with the initial conditions

$$\begin{aligned} u(x, 0, 0) &= 1 + \frac{[1 - \exp(Re(x-1))]}{[1 - \exp(-Re)]}, \\ u(0, y, 0) &= 1 + \frac{[1 - \exp(Re(y-1))]}{[1 - \exp(-Re)]}, \\ u &= 1 \qquad \text{otherwise}, \end{aligned} \tag{24}$$

where

$$Re = 1/\mu, \tag{25}$$

and the boundary conditions

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

$$u(0, y, t) = 1 + \frac{[1 - \exp(Re(y-1))]}{[1 - \exp(-Re)]},$$
$$u(x, 1, t) = 1, \tag{26}$$
$$u(1, y, t) = 1.$$

This problem has the steady-state solution

$$u = 1 + \frac{[1 - \exp(Re(x-1))][1 - \exp(Re(y-1))]}{[1 - \exp(-Re)]^2}. \tag{27}$$

MacCormack's method was used to integrate Eq. (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 ($\bar{u}$)

$$\bar{u} = fu + (1-f)(4 - x - y)/2, \qquad 0 \leqslant f \leqslant 1, \tag{28}$$

is used to calculate the driving force. Equations (5) were used to obtain grid point velocities.

Figure 11 shows the grid obtained for $Re = 5$ and $f = 0.3$. The error is calculated at the points shown in Fig. 11 and a linear interpolation is used to calculate the error at the points corresponding to the equispaced grid. The results are presented in Figs. 12–15, at each y station. The adaptive grid yields slightly higher errors in the low gradient region as in Fig. 12 and gradually progresses to much lower errors in the high gradient regions as in Fig. 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-

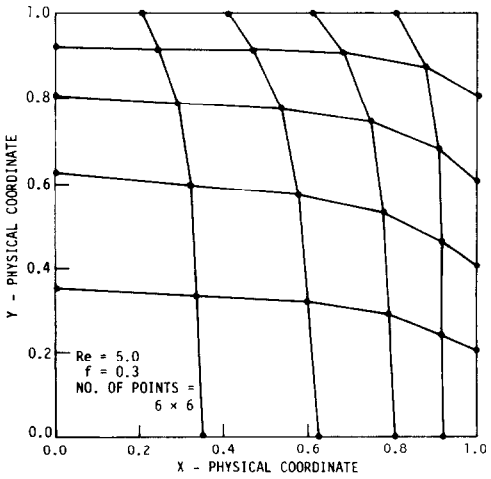

FIG. 11. Converged grid for the two-dimensional Burgers' equation.

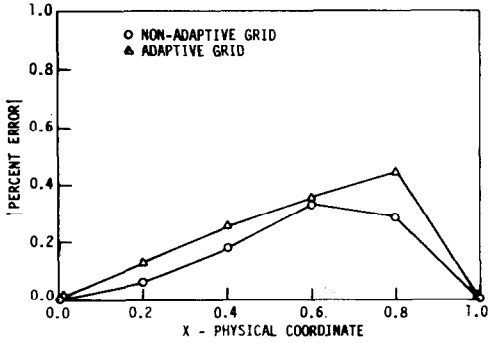

FIG. 12. Comparison of errors for the two-dimensional viscous Burgers' equation, $Re = 5.0$, $y = 0.2$.

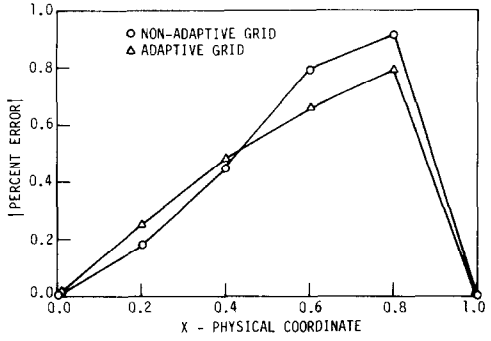

FIG. 13. Comparison of errors for the two-dimensional viscous Burgers' equation, $Re = 5.0$, $y = 0.4$.

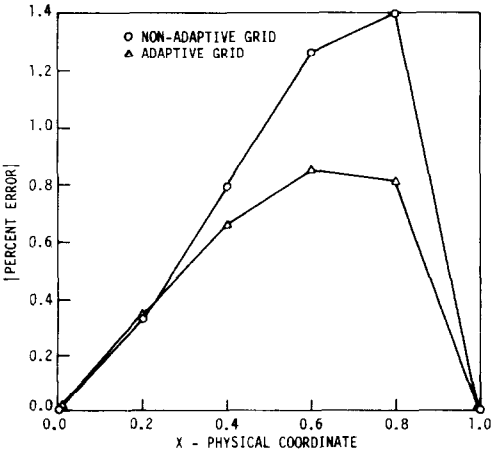

FIG. 14. Comparison of errors for the two-dimensional viscous Burgers' equation, $Re = 5.0$, $y = 0.6$.

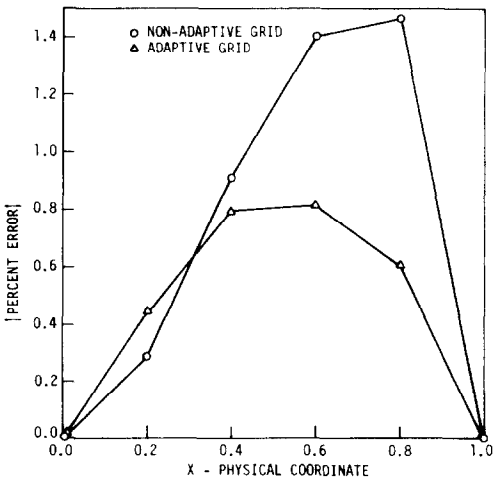

FIG. 15. Comparison of errors for the two-dimensional viscous Burgers' equation, $Re = 5.0$, $y = 0.8$.

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 Fig. 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

For an explicit method, the number of integration steps required for convergence is generally 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 take 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 additional time required

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 extra time needed for grid generation will be a much lower percentage of the total computer time required.

Concluding Remarks

A method of generating grid structure in time asymptotic problems has been presented. This required use of a coordinate transformation relating the physical plane to the computational plane where the equations governing the physical process under investigation are integrated.

Results presented show significant error reduction for the one-dimensional nonlinear viscous Burgers' equation and the two-dimensional linear viscous Burgers' equation. Error reduction was achieved by imparting a velocity to each grid point in the physical plane which depended upon both local gradient information and boundary motion. The resulting simple grid point velocity equations were then integrated in time with the governing equations yielding updated physical grid coordinates.

Extension of the present work using other measures of local error are important. In order to produce better computational grids, significant research should be done in establishing better methods for estimation of both local and global errors of solutions computed using finite difference techniques.

APPENDIX: Nomenclature

e	measure of local truncation error
E	measure of total error
f	constant used for smoothing solution
g	driving force
J	Jacobian of transformation
$K, K_1, K_2, (K)_{max}, (K_1)_{max}, (K_2)_{max}$	constants
M	number of grid points in the η direction
n	exponent
N	number of grid points in the ξ direction
r	distance between points in the computational domain
Re	Reynold's number (defined as $1/\mu$)
t	physical time
u	representative physical quantity
$\bar{u}$	smoothed form of u
$\hat{u}$	constant appearing in the steady-state solution of Burgers' equation
x	physical coordinate
y	physical coordinate

η computational coordinate
θ constant
μ viscosity term in Burgers' equation
ξ computational coordinate
τ computational time

ACKNOWLEDGMENTS

This work was sponsored by NASA under NASA Cooperative Agreement NCCI-17. The authors wish to acknowledge the encouragement and helpful suggestions provided throughout this investigation by Frank Thames and Manuel Salas at NASA Langley Research Center.

REFERENCES

1. J. F. THOMPSON, F. C. THAMES, AND C. W. MASTIN, *J. Comput. Phys.* **15** (1974), 299–319.
2. J. F. THOMPSON, F. C. THAMES, AND C. W. MASTIN, *J. Comput. Phys.* **24** (1977), 274–302.
3. F. C. THAMES, "Numerical Solution of the Incompressible Navier-Stokes Equations about Arbitrary Two-Dimensional Bodies," Ph.D. dissertation, Mississippi State University, University, Miss., 1975.
4. P. D. THOMAS AND J. F. MIDDLECOFF, *AIAA J.* **18** (June 1980), 652–656.
5. R. G. HINDMAN, P. KUTLER, AND D. A. ANDERSON, *AIAA J.* **19** (April 1981), 424–431.
6. H. A. DWYER, R. J. KEE, AND B. R. SANDERS, *AIAA J.* **18** (October 1980), 1205–1212.
7. J. OLSEN, "Subsonic and Transonic Flow over Sharp and Round Nosed Nonlifting Airfoils," Ph.D. dissertation, Ohio State University, Columbus, Ohio, 1976.
8. B. L. PIERSON AND P. KUTLER, *AIAA J.* **18** (January 1980), 49–54.
9. R. W. MACCORMACK, *AIAA PaP.* (1969), 69–354.
10. J. C. TANNEHILL, T. L. HOLST, AND J. V. RALICH, *AIAA J.* **14** (February 1976), 204–211.