JOURNAL OF COMPUTATIONAL PHYSICS 91, 32–54 (1990)

An Adaptive Grid Solution Procedure for Convection–Diffusion Problems

S. ACHARYA AND F. H. MOUKALLED

Mechanical Engineering Department, Louisiana State University, Baton Rouge, Louisiana 70803

Received October 28, 1988; July 28, 1989

A computationally efficient and stable adaptive grid solution procedure is developed for convection diffusion problems. In this method, grid refinement and adaptation is based on an equidistribution law but is only performed in regions with high error estimates that are flagged from a preliminary coarse grid solution. The equidistribution law is implicit in the grid generation procedure which requires the solution of two Poisson equations with control functions that are obtained directly from the error estimates or weighting functions at the grid points. Solution on the refined, equidistributed mesh in the flagged region is obtained with boundary conditions interpolated from the coarse grid results. Accurate solutions in both the flagged region and the coarse grid regions of the domain are obtained with a multigrid approach that requires successive solutions on the refined, equidistributed mesh in the flagged region and on the coarse mesh in the entire domain. The adaptive grid method including the multigrid calculations can be extended to several levels of refinement. The acronym LAME is coined for this procedure in view of its Local Adaptation, Multigridding, and Equidistribution features. The method is shown to be stable, computationally efficient, and accurate by applying it to three test problems and comparing with conventional calculations on a fixed curvilinear grid.

1. INTRODUCTION

Recent interest in grid generation has focussed on the development of dynamically adaptive grid systems in which the grid points move in response to the evolving solution. The grid points should move such that they are always densely clustered in regions of larger solution variations where error estimates are expected to be higher. Since the solution error is generally proportional to the product of the local grid spacing raised to some power and a measure of the local solution variation, clustering grid points in regions of large solution variation is tantamount to equidistributing a measure of the solution error over the solution domain. A dynamic procedure is necessary because the regions of large solution variations are not known a priori to the solution and evolve as the solution is driven to convergence or steady-state conditions.

A number of studies on adaptive grid solution procedures have been reported, and it is not the intent here to provide an exhaustive survey of the literature. Thompson [1] provides an excellent survey of dynamically adaptive grid systems.

0021-9991/90 $3.00

As noted in [1], all adaptive grid procedures attempt to equidistribute a measure of the error, but differ in their individual approaches.

The most popular approach has been one in which a weighting function W which is proportional to a measure of the error (typically containing the first or second derivative of the dependent variable ϕ or some combination of first and second derivatives) is calculated at all points and the mesh size Δs is adjusted so that $W \cdot \Delta s$ is nearly the same at all points in the entire domain. Approaches of this or similar kind have been reported by a number of investigators [2–11]. For the lack of a better name, this method will be referred to as the *global equidistribution approach* in this paper. Acharya and Patankar [2] have used a normalized second derivative ϕ_{yy} as the weighting function. A linear combination given by $1 + \tilde{\alpha}\,|\phi_y| + \tilde{\beta}\,|\phi_{yy}|$ was used as a measure of the weighting function by Dwyer *et al.* [3–5] and also by Gnoffo [6] (with $\tilde{\beta} = 0$). In [6], the idea of a spring analogy represented in the equation for the weighting function was introduced and was extended by Nakahashi and Deiwert [7] who introduced the notion of a torsion spring attached to each node in order to control the inclination of the grid lines. Rai and Anderson [8, 9], Greenberg [10], and Eiseman [11] have each used the idea of moving the grid points under the influence of forcing or weighting functions that either attract or repel grid points relative to each other. Thus points with forcing (or weighting) functions greater than a specified average value attract each other, and those with values less than the average value repel each other.

A second approach (called the *variational approach* here) is to use a variational principle in which the grid generation system can be obtained from the Euler's equation for the minimization of an integral whose integrand is proportional to a measure of the error. As shown in [1], this minimization process is equivalent to the solution of a Poisson equation with the inhomogeneous terms proportional to a measure of the error. Brackbill and Saltzman [12, 13], Yanenko *et al.* [14], and Bell and Shubin [15] have each used the variational principle for grid adaptation.

The two approaches outlined above generally involve the movement of all the grid points in the domain. These approaches are associated with problems of excessive grid skewness, oscillations in the solution, etc. A third approach, which avoids some of these problems is one in which the mesh is refined only in the region where the error estimate exceeds a critical value. This method is termed the *local mesh refinement approach.* Examples of such an approach are due to Berger and Jameson [16], Phillips and Schmidt [17], and Caruso *et al.* [18]. Typically grid points with higher error levels are flagged and the flagged region is expanded in order to make it rectangular. The mesh size in the rectangular flagged region is typically halved and calculations repeated only in the flagged, finer grid region. In [18], an additional step similar to that in a *multi-grid method* was performed by repeating the calculations in the original coarse grid but with correction terms added to the coarse grid equations in the flagged region so that the solution in that region matches the corresponding fine grid solutions. There are two major disadvantages of the studies reported under this approach. The first is that the flagged region is generally subdivided uniformly without consideration of the relative

magnitudes of the errors at the flagged points and therefore the mesh in the flagged region is less than optimal. The second disadvantage is that a non-rectangular flagged region (in the computational space) is generally accounted for by including non-flagged points so as to generate a rectangular domain or by using overlapping rectangular domains. Both these practices are computationally inefficient.

In this paper the development of a stable, computationally efficient adaptive grid solution procedure is sought. Of the three approaches described so far, the method developed in this paper is probably best described by the local mesh refinement approach, but unlike earlier studies, the present method is capable of efficiently dealing with non-rectangular flagged regions, and more importantly, grid refinement in each such flagged region is based on equidistribution concepts such that a near-optimal grid in each flagged region is obtained. In addition, multigrid concepts have been built into the adaptive grid procedure so that the solution accuracy in the entire domain and in the refined flagged region are both significantly improved. The resulting adaptive grid procedure is stable and computationally efficient and details of this procedure are described next.

In the discussion that follows, the discretization procedure on a curvilinear grid is first described followed by the details of the adaptive grid technique, and finally the results of the adaptive grid solution procedure on a number of test problems are presented.

2. Discretized Conservation Equations

The numerical solution on a curvilinear grid involves three steps: grid generation, discretization of the conservation equations, and solution of the discretized equations. Since many of these details can be found elsewhere [19, 20], only the important issues are briefly described here.

Grid Generation

Generating a grid in an arbitrary physical domain involves a coordinate transformation from the physical space (x, y) to the computational space (ξ, η) (see Fig. 1). This is done here by solving a system of Poisson equations [20],

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = P_1 \tag{1}$$

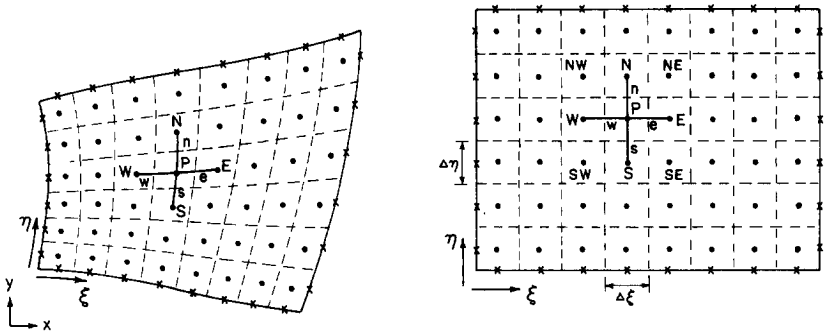
$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = P_2, \tag{2}$$

where

$$\alpha = x_\eta^2 + y_\eta^2, \qquad \beta = x_\xi x_\eta + y_\xi y_\eta, \qquad \gamma = x_\xi^2 + y_\xi^2 \tag{3}$$

and

$$P_1 = -J^2[x_\xi P + x_\eta Q], \qquad P_2 = -J^2[y_\xi P + y_\eta Q]. \tag{4}$$



(a) Physical Space (b) Computational Space

FIG. 1. Schematic of curvilinear coordinate system.

In the above equations, J is the Jacobian ($=x_\xi y_\eta - x_\eta y_\xi$) and P and Q are control functions which can be chosen in order to provide a denser distribution of points in certain regions. Anderson and Steinbrenner [21] have shown that with proper choices for P and Q that relate them to the local error estimates or weighting functions, Eq. (1) and (2) can be interpreted as an equidistribution law. In this paper, advantage is taken of this fact and Eq. (1) and (2) are used for generating the initial curvilinear grid and also the adaptive grid in each flagged region. Additional details are given in the next section.

Central differences are used to discretize Eq. (1) and (2). The resulting system of algebraic equations are solved iteratively by a line by line tri-diagonal matrix or Thomas algorithm [22]. The resulting solution gives x, y values at uniform ξ, η values, and all metric quantities can be calculated from this solution.

Discretization of the Conservation Equations

The conservation equations for the convection–diffusion of a transport variable ϕ in curvilinear coordinates can be expressed as

$$\{\rho G_1 \phi - (\Gamma/J)(\alpha\phi_\xi - \beta\phi_\eta)\}_\xi + \{\rho G_2 \phi - (\Gamma/J)(\gamma\phi_\eta - \beta\phi_\xi)\}_\eta = SJ, \tag{5}$$

where S is the source term and G_1 and G_2 are the contravariant velocity components defined by

$$G_1 = uy_\eta - vx_\eta, \qquad G_2 = vx_\xi - uy_\xi \tag{6}$$

and ρ and Γ denote the density and diffusion coefficient of the fluid, respectively. The task of the discretization process is to approximate the differential equation by algebraic equations at the grid points. To this end, a control-volume based finite difference procedure is adopted in this study, in which control volumes are first defined around each grid point, as shown in Fig. 1. Equation (5) is then expressed in an integral form over each control volume consisting typically of a grid point P, with E, W, N, and S (east, west, north, and south) neighboring grid points and

corresponding e, w, n, and s control volume faces. This leads to an integral balance equation of the form

$$[\rho G_1 \phi - (\Gamma/J)(\alpha\phi_\xi - \beta\phi_\eta)]_w^e \Delta\eta + [\rho G_2 \phi - (\Gamma/J)(\gamma\phi_\eta - \beta\phi_\xi)]_s^n \Delta\xi = S_P J \Delta\eta \Delta\xi \quad (7)$$

where $[\cdots]_w^e$ denotes $[\cdots]_e - [\cdots]_w$. The terms in the parenthesis ϕ, ϕ_ξ, and ϕ_η are expressed at the interfaces w, e, s, and n in Eq. (7), but since ϕ is stored at the grid points, the interface values have to be expressed in terms of the grid point values. Therefore profile approximations are necessary. In this paper, the Power-Law profile has been chosen [23], since it has been shown to be more accurate than either the central-difference, upwind, or hybrid schemes [24]. The resulting algebraic equation has the form

$$a_P \phi_P = a_E \phi_E + a_W \phi_W + a_N \phi_N + a_S \phi_S + b_S + b_{\text{no}}, \quad (8)$$

where a_P, a_E, a_W, a_N, and a_S are convection–diffusion coefficients, b_S is the source term contribution, and b_{no} is the contribution due to non-orthogonality of the grid. The expressions for these coefficients, and additional details are given elsewhere [20].

Solution of the Discretized Equations

Equation (8) represents a penta-diagonal system of equations and is solved here by a line by line Thomas algorithm [22]. This line by line algorithm is similar, in concept, to the ADI method and is described in greater detail in [24].

3. Adaptive Grid Solution Procedure

In the adaptive grid procedure developed in this paper, the grid is refined in flagged regions where the error levels or weighting functions exceed a specified threshold value. Thus, all or nearly all of the grid points in the domain are not forced to move each time the grid adaptation is performed. This reduces the problem of oscillations between the grid and the solution that characterizes some of the global equidistribution methods [2–5]. In this regard, the present method may be considered to fall under the local mesh refinement category. However, as noted earlier, the method developed in this paper differs from earlier studies under this category and embodies a number of major contributions. The first is that it can handle an arbitrary cluster of flagged points that do not necessarily yield a rectangular flagged region in the computational space. Second, in each flagged region the mesh refinement is not done uniformly as in earlier local mesh refinement studies [16–18], but by an equidistribution law that provides denser clustering in those areas of the flagged region where the error levels or weighting functions are relatively higher. In addition, since the solution accuracy in the flagged region depends on the accuracy of the interpolated boundary condition, it is also important to have an accurate solution outside the flagged region. To this end, a multi-

grid approach is incorporated in the present grid adaptation strategy, by which, the solution errors in both the flagged domain and the entire domain are successively reduced.

The solution-adaptive grid procedure developed here is therefore characterized by Local Adaptation with Multigridding and Equidistribution concepts (LAME). The major steps of this procedure are flagging of points to generate flagged regions (that are not necessarily rectangular), generating a finer mesh in the flagged regions using an equidistribution law, interpolation of boundary conditions along the boundaries of the flagged region, and a multi-grid-type calculation between the flagged region and the entire domain. These details are described next.

Defining Flagged Regions

The solution process is initiated by generating a relatively coarse mesh in the domain by solving the set of Poisson equations (Eq. (13) and (14) or Eq. (1) and (2)) with zero or specified control functions and performing a pre-assigned number of iterative calculations on this grid. An error estimate or weighting function W is then calculated at each grid point. For this purpose W is defined as

$$W = \alpha_1 J\,|\nabla^2\phi| + \alpha_2(\nabla\phi)^2 + \alpha_3 J\,|\nabla\xi\cdot\nabla\eta| + \alpha_4 J(\nabla\xi^2 + \nabla\eta^2). \tag{9}$$

The first term represents the rate of change of the gradient of the dependent variable ($\nabla^2\phi$), the second term is proportional to the gradient of the dependent variable ($\nabla\phi$), the third term is the local orthogonality of the grid ($\nabla\xi\cdot\nabla\eta$), and the last term is a measure of the grid smoothness. Each term is of the same order of magnitude, and the relative importance of the various terms is controlled through the constants α_1, α_2, α_3, and α_4.

To flag points, a normalized weighting function $\bar{W}$ is defined as

$$\bar{W} = (1 + W)/(1 + W_{\max}) \tag{10}$$

and points are flagged if the normalized weighting function $\bar{W}$ is greater than a preassigned value. A flagged region is generated by identifying a cluster of contiguous points. To create the cluster, a flagged point is initially chosen, and flagged points are added to the cluster if they are the neighbors of a flagged point already in the cluster. The cluster is closed if no new flagged points can be added to it. A new cluster or flagged region is then generated with another flagged point (not yet enclosed in a cluster) acting as a nucleus or starting point for the new cluster. By this means, clusters of flagged points or flagged regions are generated until all the flagged points are enclosed in one cluster or another.

The boundaries of the flagged region, in the computational space, are the control volume faces of the outermost string of points in the cluster (Fig. 2). Therefore, unlike earlier studies, no constraint requiring the flagged regions to be rectangular (or consisting of overlapping rectangles) in the computational space is imposed. This is potentially advantageous, particularly from the viewpoint of computational efficiency.

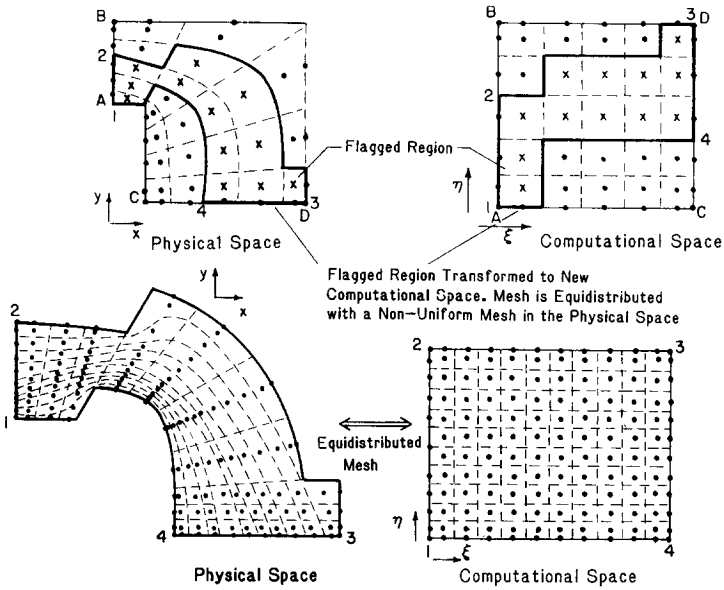


FIG. 2. A typical flagged region.

Grid Generation in the Flagged Region Using an Equidistribution Scheme

Once flagged regions have been defined in the computational domain, the next step is to generate a finer mesh in the flagged region using an equidistribution scheme such that areas within the flagged region that are associated with higher error levels or weighting functions have the densest grid point clustering. Thus a more optimal mesh in the flagged region will be obtained compared to the mesh obtained by simply reducing the mesh size everywhere, say by half. The latter practice has been adopted in earlier local mesh refinement studies [16–18].

To obtain a refined grid in the flagged region using an equidistribution scheme, the ideas of Anderson and Steinbrenner [21] are used. In [21] it is shown that the conventional grid generation procedure of solving two Poission equations (Eq. (1) and (2)) can be interpreted as an equidistribution scheme, if the control functions P and Q are related to local error levels or weighting functions. This approach of using control functions P and Q in the Poisson equations to force grid movement in order to obtain an equidistributed grid has also been established from variational principles [25, 26]. Thus the same grid generation routine that is used in obtaining the initial curvilinear grid in the entire domain can be used to calculate the equidistributed grid in each flagged region. This practice is computationally efficient.

The relationship between the weighting functions W (or the error levels) and the control functions P and Q in Eq. (1) and (2) can be most easily derived by considering the one-dimensional equidistribution law along the constant-η line, i.e.,

$$(s_1)_\xi W = f_1(\eta), \tag{11}$$

where s_1 is the arc length along a line of constant η, $f_1(\eta)$ is a constant for a given curve, and W is the weighting function. Differentiating Eq. (11) with respect to ξ, the following equation is obtained:

$$(s_1)_{\xi\xi} + [(W)_\xi / W](s_1)_\xi = 0. \tag{12}$$

To relate the above equidistribution law to Eq. (1), the control functions are redefined as $\tilde{\phi} = J^2 P/\alpha$ and $\tilde{\psi} = J^2 Q/\gamma$ and substituted into Eq. (1) and (2), resulting in

$$\alpha(x_{\xi\xi} + \tilde{\phi} x_\xi) - 2\beta x_{\xi\eta} + \gamma(x_{\eta\eta} + \tilde{\psi} x_\eta) = 0 \tag{13}$$

$$\alpha(y_{\xi\xi} + \tilde{\phi} y_\xi) - 2\beta y_{\xi\eta} + \gamma(y_{\eta\eta} + \tilde{\psi} y_\eta) = 0. \tag{14}$$

Eliminating $\tilde{\psi}$ between the above two equations, and assuming orthogonality and zero curvature the following equation for the distance along a constant η arc is obtained:

$$(s_1)_{\xi\xi} + \tilde{\phi}(s_1)_\xi = 0. \tag{15}$$

If the orthogonality and zero curvature constraints are removed then a similar equation is obtained, i.e.,

$$(s_1)_{\xi\xi} + \phi_1(s_1)_\xi = 0, \tag{16}$$

where

$$\phi_1 = \tilde{\phi} - [(\theta_1)_\xi - 2(\theta_2)_\xi] \cot \theta_3 - (s_1)_\xi (\theta_2)_\eta \sin \theta_3 / (s_2)_\xi \tag{17}$$

and θ_1 and θ_2 are the slopes of the constant η and ξ curves and θ_3 is the angle of intersection ($\theta_2 = \theta_1 + \theta_3$).

Comparing Eq. (16) and (12) leads to a relationship between the function ϕ_1 and the weighting function W, and a similar relationship can be derived between ψ_1 and W. These relationships are

$$\phi_1 = (W)_\xi / W, \qquad \psi_1 = (W)_\eta / W. \tag{18}$$

Thus the conventional elliptic grid generation procedure [19] can be interpreted as an equidistribution law if the control functions $\tilde{\phi}$ and $\tilde{\psi}$ (in Eq. (13) and (14)) are obtained from Eq. (17) and (18). This approach is incorporated in the present work.

The expression for the weighting function W, given in Eq. (9), consists of four terms, and as explained earlier, these represent the gradient ($\nabla\phi$), its rate of change ($\nabla^2\phi$), the local orthogonality ($\nabla\xi \cdot \nabla\eta$) and the grid smoothness ($\nabla\xi^2 + \nabla\eta^2$). The relative importance of the various terms can be controlled through the constants α_1, α_2, α_3, and α_4. In this paper, α_3 and α_4 have been set equal to zero.

To generate the refined grid in each flagged region, the number of grid points along the boundaries of the flagged region are increased. In this paper, they have

been doubled. The $\xi = \xi_{\min}$, $\xi_{\max}$ and $\eta = \eta_{\min}$, $\eta_{\max}$ boundaries of the flagged region are then chosen. Since the flagged region is not rectangular in the original ξ, η computational space, the choice of $\xi_{\min}$, $\xi_{\max}$ and $\eta_{\min}$, $\eta_{\max}$ is not along the orginal ξ and η coordinate directions, respectively. Rather the points along the boundaries of the flagged region are divided into four parts and each part is assigned to form a new computational boundary, i.e., the $\xi_{\min}$, $\eta_{\min}$, $\xi_{\max}$, $\eta_{\max}$ boundaries, respectively. This is illustrated in Fig. 2, where the boundaries of the flagged region (dark lines) are divided into four parts with 1–2 and 4–3 assigned as $\eta_{\min}$ and $\eta_{\max}$ and 1–4 and 2–3 as $\xi_{\min}$ and $\xi_{\max}$. With the new boundary grid points assigned equally to each of the four boundaries of the flagged region, a preliminary grid is guessed or generated in the flagged region and the values of the weighting functions W and the control functions ϕ_1 and ψ_1 are interpolated onto this grid from the corresponding coarse grid values. Equations (13) and (14) are then solved, with the ϕ_1 and ψ_1 values interpolated as stated above, and a new equidistributed grid is generated in the flagged region. This procedure can be extended further and taken to convergence by again interpolating fresh ϕ_1 and ψ_1 values on to the new equidistributed grid, and solving Eq. (13) and (14). The procedure is repeated until grid points undergo no further change in position. As a final step, boundary grid point locations are adjusted so that the grid is orthogonal along the boundaries of the flagged region. This is done by requiring the distance l between the first interior grid point x_i, y_i and the boundary point $x(s)$, $y(s)$ to be a minimum, i.e.,

$$\partial l/\partial s = (x_i - x(s))(\partial x/\partial s) + (y_i - y(s))(\partial y/\partial s) = 0. \tag{19}$$

Piecewise cubic spline profiles are used to represent $x(s)$ and $y(s)$ as

$$x(s) = a_0 + a_1 s + a_2 s^2 + a_3 s^3, \qquad y(s) = b_0 + b_1 s + b_2 s^2 + b_3 s^3. \tag{20}$$

The coefficients $a_0 \cdots a_3$ and $b_0 \cdots b_3$ are obtained from the original grid point values along the boundary. Equations (19) and (20) are solved using Newton's method to obtain the boundary grid point locations.

The above procedure for generating an equidistributed grid is repeated in each flagged region (whether they are multiply connected or not). Only after the refined grid in each flagged region is completed, is the solution process initiated.

Boundary Condition for the Flagged Region

In order to obtain a solution in the flagged region, boundary conditions have to be interpolated from the coarse grid solution along the boundaries of the flagged region. The values at the corner of the coarse grid control volume faces along the boundaries of the flagged region are first determined as the weighted average of the four neighboring coarse grid points. Linear interpolation is then used between the coarse grid corner values to calculate the values at the fine grid points along the boundary of the flagged region.

Correction Terms for the Coarse Grid Equations in Flagged Regions

Since the fine grid solution in the flagged region is obtained with boundary conditions interpolated from the coarse grid solution, the latter should be reasonably accurate. To this end, a multi-grid type approach is adopted in this paper by which consecutive calculations are performed on the coarse grid and fine grid with information exchange between the two solutions (called restriction and prolongation operations in a multi-grid method) in each cycle of calculation. In this paper, this approach is incorporated by first obtaining a solution on the coarse grid, followed by the fine grid solution in the flagged regions with boundary conditions interpolated from the coarse grid solution. The coarse grid equations are again solved but with corrected coarse grid equations in the flagged region such that the resulting coarse grid solution in the flagged region matches the corresponding fine grid solution. New boundary conditions are interpolated for the flagged regions from the improved coarse grid solution and the solution in the flagged regions is again obtained. These consecutive calculations are repeated until convergence within a specified tolerance.

To derive the corrected coarse grid equations in the flagged region, the conservative property is invoked by which the total flux on the coarse grid must equal the total flux on the fine grid. Denoting the coarse grid values with a superscript 1 and the fine grid values by a superscript 0, this conservative property on the coarse mesh is expressed by

$$\begin{aligned}&[\rho^1 G_1^1 \phi^1 - (\Gamma^1/J^1)(\alpha^1 \phi_\xi^1 - \beta^1 \phi_\eta^1)]_\xi + [\rho^1 G_2^1 \phi^1 - (\Gamma^1/J^1)(\gamma^1 \phi_\eta^1 - \beta^1 \phi_\xi^1)]_\eta \\ &\quad = [\rho^0 G_1^0 \phi^0 - (\Gamma^0/J^1)(\alpha^1 \phi_\xi^0 - \beta^1 \phi_\eta^0)]_\xi \\ &\qquad + [\rho^0 G_2^0 \phi^0 - (\Gamma^0/J^1)(\gamma^1 \phi_\eta^0 - \beta^1 \phi_\xi^0)]_\eta. \end{aligned} \tag{21}$$

Assuming the same physical properties ($\rho^1 = \rho^0$, $\Gamma^1 = \Gamma^0$) and conservation of mass ($G_1^1 = G_1^0$, $G_2^1 = G_2^0$) on both grids, the above equation reduces to

$$[Lx^1 \phi^1]_\xi + [Ly^1 \phi^1]_\eta = [Lx^1 \phi^0]_\xi + [Ly^1 \phi^0]_\eta, \tag{22}$$

where

$$Lx\phi = \rho G_1 \phi - (\Gamma/J)(\alpha \phi_\xi - \beta \phi_\eta), \qquad Ly\phi = \rho G_2 \phi - (\Gamma/J)(\gamma \phi_\eta - \beta \phi_\xi). \tag{23}$$

Equation (22) is the coarse grid correction equation in the flagged region. Although derived by requiring fluxes to be conserved on both grids, Eq. (22) can be directly derived by requiring the coarse grid values (ϕ^1) to be equal to the fine grid values (ϕ^0) in the flagged region. This requirement directly leads to Eq. (22).

Comparing Eq. (5) and (22), it can be seen that the left-hand sides of the two equations are the same, but the right-hand sides are different. In Eq. (22), the right-hand side may be viewed as a correction term so that ϕ^1 will be equal to ϕ^0 in the flagged region. Thus, after the fine grid solution ϕ^0 is obtained in the flagged region, the coarse grid calculations are done by solving Eq. (5) outside the flagged regions

and by solving Eq. (22) in the flagged regions. This will lead to more accurate coarse grid solutions.

The Generalized LAME Procedure

The description, so far, has been given with one level of refinement in mind, but, in practice, the method is not limited to one level of refinement and generalization to multiple levels of refinement is straightforward. If the solution on the coarsest grid is denoted by V_1, and the solution on successively finer grids in the flagged regions by V_0, V_{-1}, V_{-2}, ..., V_{-i+1}, $V_{-i...}$, V_{-k}, then multigrid type calculations can be performed by iterating from the coarsest to the finest grid and then back again. Each level of refinement has its own flagged region (say Ω_{-i}) that typically will be embedded in the flagged region corresponding to the previous refinement (Ω_{-i+1}). This generalized LAME procedure is described by the following steps:

1. Define a preliminary coarse grid (Ω_1) in curvilinear coordinates in the domain, and obtain a solution V_1 on it.

2. For the first level of refinement ($i=0$), flag grid points if normalized weighting functions exceed a specified value. For higher levels of refinement ($-i$) flag grid points with normalized weighting functions higher than those specified in the previous level of refinement ($-i+1$). Thus flagged regions (Ω_{-i}) are generated at any level of refinement, and typically, Ω_{-i} is embedded in Ω_{-i+1}.

3. Generate a finer mesh in Ω_{-i} based on an equidistribution scheme in each flagged region.

4. Interpolate boundary conditions along the boundaries of each flagged region from the available solution on the previous mesh (V_{-i+1}).

5. Obtain the solution V_{-i} in each flagged region Ω_{-i}.

6. Calculate correction terms in Ω_{-i} for the equations on the previous mesh (Ω_{-i+1}). Solve the new set of equations in Ω_{-i+1} to obtain an improved ($-i+1$) level solution, V_{-i+1}.

7. Repeat steps 4–6 until a specified level of convergence is obtained.

8. Advance to the next level of refinement and repeat steps 2–7.

9. Proceed until the solution in the finest desired level of refinement V_{-k} is obtained.

4. Results and Discussion

The performance of the adaptive grid solution procedure is examined by solving three typical problems. Adaptive grid calculations are initiated on a preliminary 11×11 or 12×12 coarse grid as explained earlier. Therefore adaptive grid results are compared with a conventional 11×11 or 12×12 curvilinear grid calculation. Since adaptive grid calculations require more work (or cpu time) per step, it is also

appropriate to compare the adaptive grid results with the conventional calculation on a uniformly finer curvilinear grid that takes the same amount of cpu time. For all calculations, only one level of refinement is performed. To make a meaningful comparison, both solutions are driven to the same level of convergence.

Test Problem 1—Heat Conduction with Discontinuous Boundary Temperature

The first test problem is that of two-dimensional heat conduction in a rectangular plate with a step change in temperature along the upper boundary. The physical situation is depicted in Fig. 3. The governing differential equations and boundary conditions are

$$T_{xx} + T_{yy} = 0 \tag{24}$$

$$T(0, y) = T(1, y) = T(x, 0) = 0 \tag{25}$$

$$T(x, 1) = 1 \quad \text{for} \quad 0 < x \leqslant \tfrac{1}{2}; \qquad T(x, 1) = -1 \quad \text{for} \quad \tfrac{1}{2} < x < 1. \tag{26}$$

An analytical solution to Eq. (24)–(26) is given by

$$T(x, y) = \sum_{k=1}^{\infty} (2/k\pi)[1 + \cos(k\pi) - 2\cos(k\pi/2)] \sin k\pi x \times \sinh(k\pi y)/\sinh(k\pi). \tag{27}$$

Since the largest temperature gradients are expected near the upper boundary, and particularly in the vicinity of $(\frac{1}{2}, 1)$, the flagged region is also expected to be at the top. Figure 4 shows the flagged region, and the mesh generated using the equidistribution scheme described earlier. As expected, the finest mesh is obtained in the vicinity of $(\frac{1}{2}, 1)$, $(0, 1)$, and $(1, 1)$. As may be seen, the mesh in the flagged region appears to have the desirable features.

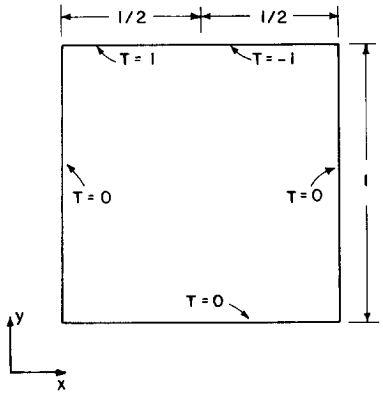


FIG. 3. Physical domain and boundary condition for Test Problem 1 (heat conduction with discontinuous boundary temperature).

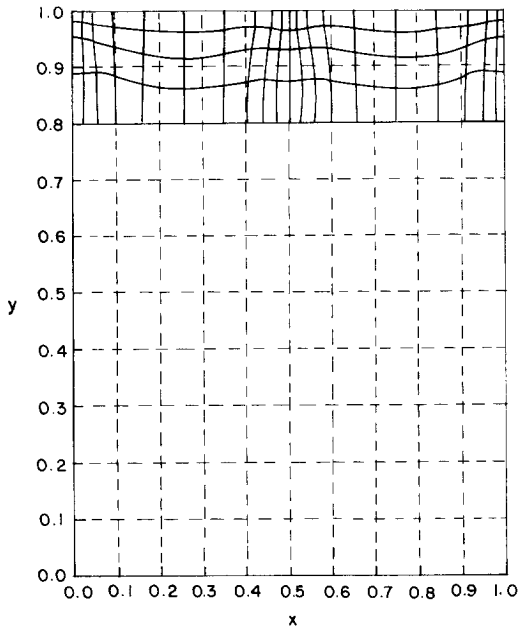


FIG. 4. Preliminary 12×12 coarse grid (dashed lines) and refined grid in flagged region (solid lines) for Test Problem 1. (Note that all lines denote control volume faces. Grid points are at the center of each control volume and along the boundaries.)

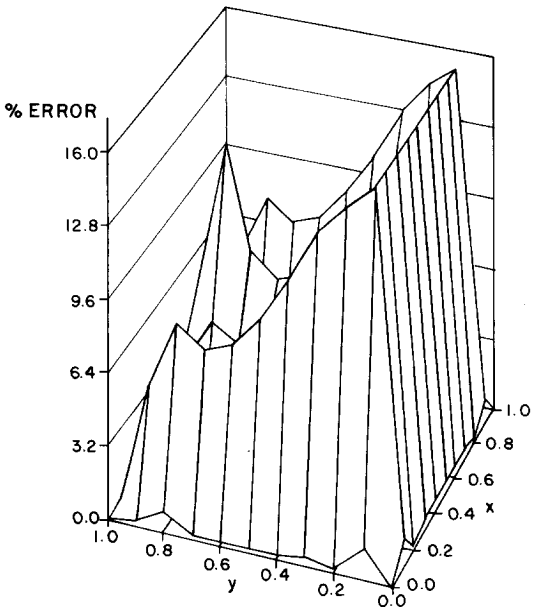


FIG. 5. Percentage error with a conventional calculation on a 12×12 uniform grid for Test Problem 1.

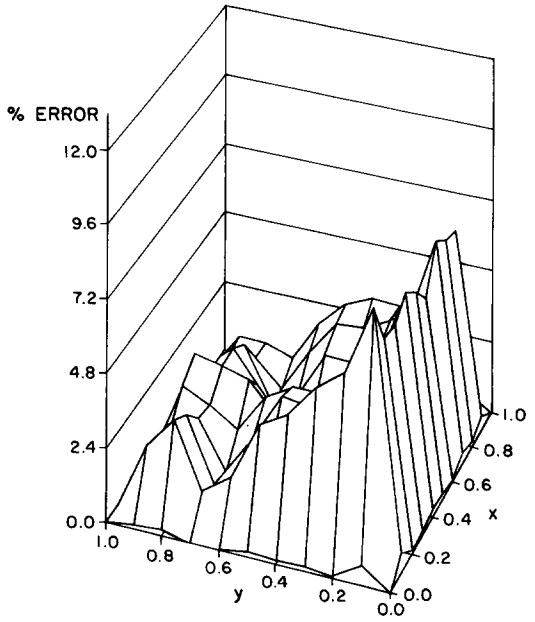


FIG. 6. Percentage error with an adaptive grid procedure initiated on a 12×12 uniform grid for Test Problem 1.

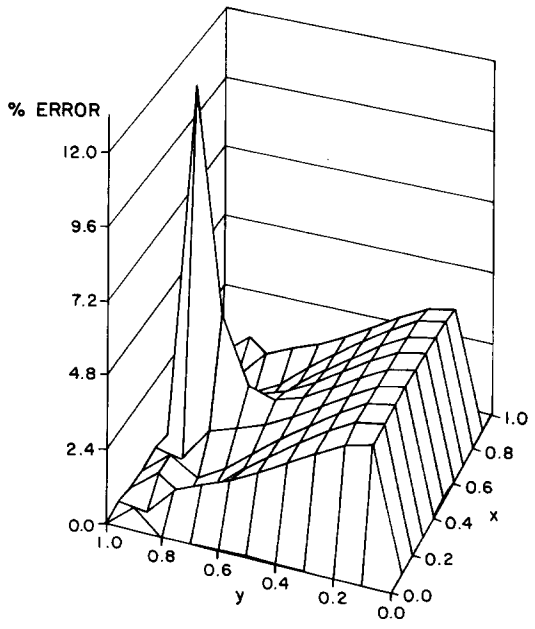


FIG. 7. Percentage error with a conventional calculation on a 20×20 uniform grid for Test Problem 1.

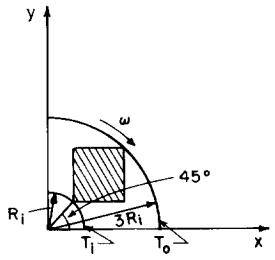


FIG. 8. Physical domain and boundary condition for Test Problem 2 (conduction in a rotating hollow cylinder).

Figures 5–7 show the percentage error in the solutions on a conventional 12×12 grid (Fig. 5), on an adaptive or equidistributed grid generated from an initial 12×12 coarse grid (Fig. 6), and on a finer 20×20 conventional grid whose solution requires the same effort as the adaptive grid calculation (Fig. 7). Conventional solution on a 12×12 mesh has errors ranging from 5 to 15% over most of the domain. The adaptive grid calculations dramatically reduce the error levels to values typically in the 2 to 5% range. The conventional calculations on a finer 20×20 grid requiring the same effort as the adaptive grid procedure has error levels as high as 11% (near $(\frac{1}{2}, 1)$, where temperature gradients are large.

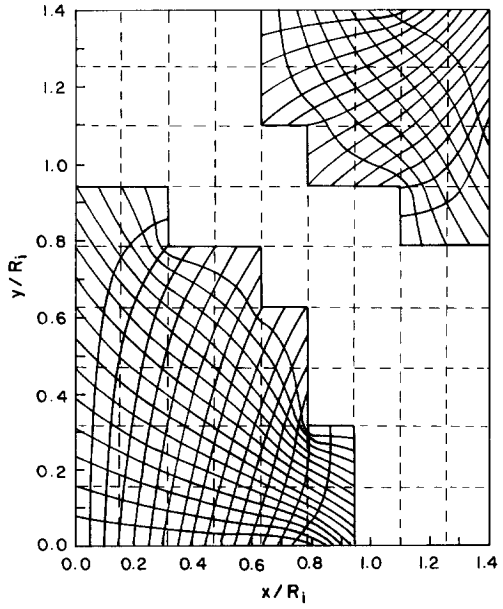


FIG. 9. Preliminary 11×11 coarse grid (dashed lines) and refined grid in the flagged region (solid lines) for Test Problem 2 (see additional comments in legend of Fig. 4).

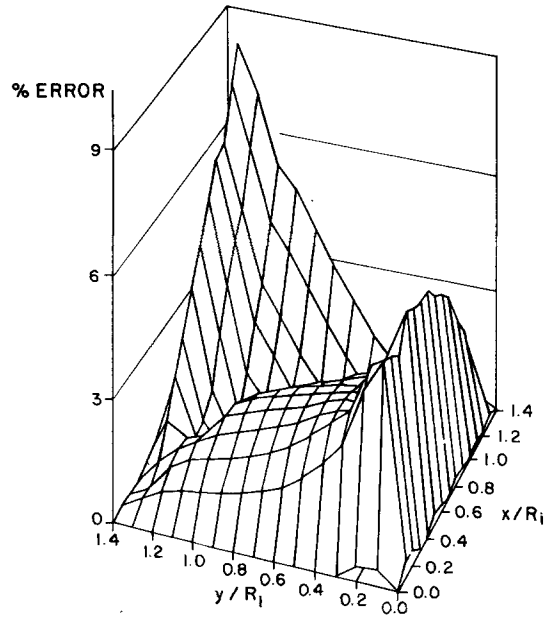


FIG. 10. Percentage error with a conventional calculation on a 11×11 uniform grid for Test Problem 2, Pe = 100.

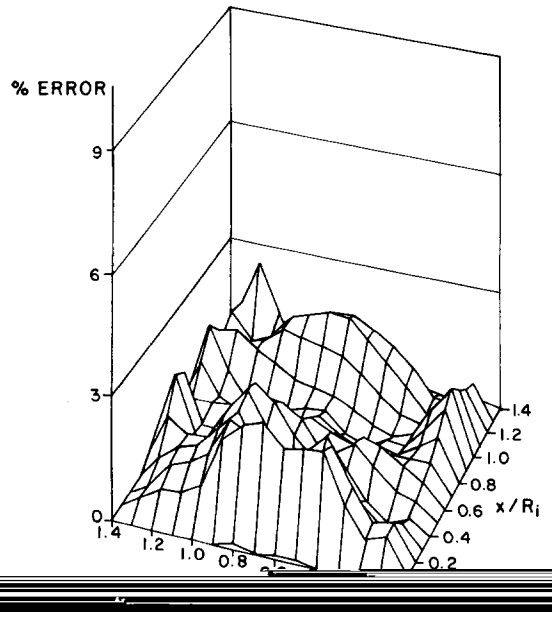


FIG. 11. Percentage error with an adaptive grid procedure initiated on a 11×11 uniform grid for Test Problem 2, Pe = 100.

It should be pointed out that in the adaptive grid calculation the mesh in the middle and lower parts of the domain is the original 12×12 uniform mesh on which the calculations are initiated and this mesh is considerably coarser than the conventional 20×20 mesh. Yet, the errors in the nonflagged region in the adaptive grid procedure are considerably lower than the corresponding errors with the conventional calculation on the 12×12 grid and comparable to the errors in the solution in the finer 20×20 grid. This demonstrates the power of the multigrid type calculation incorporated in the present work.

Test Problem 2—Conduction in a Rotating Hollow Cylinder

Radial heat conduction in a rotating hollow cylinder is a commonly studied test problem used to test numerical schemes for convection–diffusion problems [25, 26]. A schematic of the physical situation is shown in Fig. 8. The angular speed is ω and the temperatures of the inner and outer surfaces are T_i and T_0. The density ρ, specific heat c_p, and thermal conductivity k of the cylinder are assumed to be constant.

In polar coordinates, the problem is one-dimensional and an analytical solution is available. In a cartesian domain, shown shaded in Fig. 8, the problem is two-dimensional and is described by the following dimensionless convection–diffusion equation

$$U\phi_X + V\phi_Y = (\phi_{XX} + \phi_{YY})/Pe, \tag{28}$$

where

$$\begin{aligned} &\phi = (T - T_0)/(T_i - T_0), \qquad U = u/(\omega R_i), \\ &V = v/(\omega R_i), \qquad Pe = \rho\omega^2 R_i^2/(k/c_p) \end{aligned} \tag{29}$$

and

$$X = x/R_i, \qquad Y = y/R_i. \tag{30}$$

Adaptive grid calculations are initiated on a 11×11 grid for a Peclet number Pe of 100, and results are compared with the analytic solution given by

$$U = 2Y, \qquad V = -2X, \qquad \phi = 1 - \mathrm{Ln}(X^2 + Y^2)/2 \,\mathrm{Ln}\, 3. \tag{31}$$

Figure 9 shows the original 11×11 coarse mesh (9×9 control volumes), the flagged regions at the two corners and the refined, equidistributed mesh in each flagged region. Since the refined mesh is generated by solving a set of elliptic equations which are inherently characterized by a smoothing effect, the refined mesh near the staircase boundary does not exhibit the staircase profile. As a result, some of the near boundary control volumes along the staircase boundaries are coarser than is desirable. A remedy to this problem is to join the outermost string of flagged points in a cluster by straight lines and to use this as the boundary of the flagged region. This remedy is adopted in the next problem.

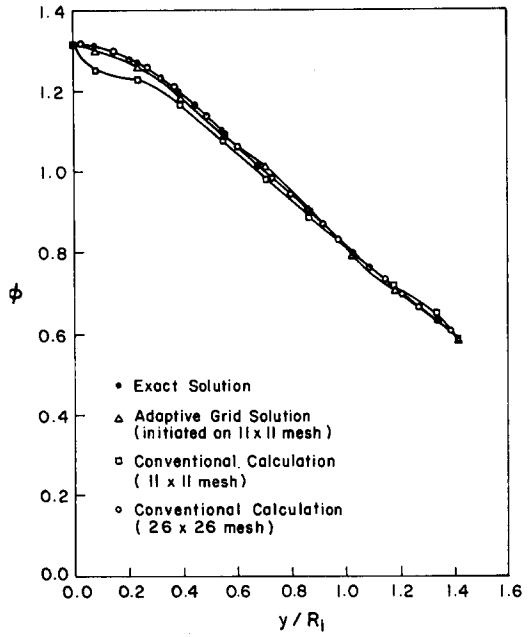


FIG. 12. Centerline profile for ϕ in Test Problem 2, Pe = 100.

Figures 10 and 11 show the percentage error in the solution obtained with a conventional calculation on a 11×11 grid and with an adaptive grid solution initiated on a 11×11 grid. Compared to the conventional calculation where error levels are as high as 8%, the maximum error in the adaptive grid calculation is about 2.5%. A conventional calculation on a 26×26 grid takes about the same cpu time as the adaptive grid solutions and error levels of the two solutions are comparable. This is seen in Fig. 12, where the centerline temperature profile is plotted. The exact solution, the adaptive grid solution, and the conventional solution on a 26×26 grid are nearly identical to each other.

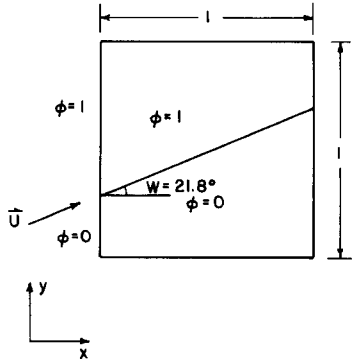


FIG. 13. Physical domain and boundary condition for Test Problem 3 (transport of a step profile in a uniform velocity field).

Test Problem 3—Transport of a Step Profile in a Uniform Velocity Field

This is another typical problem employed by a number of researchers to test various numerical schemes developed to reduce false diffusion errors that become significant in velocity fields inclined with respect to the mesh and at high Peclet numbers [25, 26]. Therefore this problem is a good test of the ability of the adaptive grid procedure to reduce false diffusion errors that are proportional to the mesh size and the angle of the velocity vector with respect to the grid lines in the physical space.

The physical situation is shown in Fig. 13 and is governed by the equation

$$(u\phi)_x + (v\phi)_y = 0, \tag{32}$$

where ϕ is the dependent variable and u and v are the components of the uniform velocity vector **u**, which in this problem is assumed to be at an angle of 21.8° with respect to the horizontal. At $x=0$, a step profile in ϕ is assumed as shown in Fig. 13, and since diffusion has been assumed to be zero in Eq. (32) and the velocity field is uniform, this step profile must be convected at an angle of 21.8° to the horizontal, with the exact solution being $\phi=1$ above the 21.8° line and $\phi=0$ below it.

The largest gradients, and therefore the flagged region, is expected to be in the vicinity of the region where the profile has a step in it. This expectation is confirmed

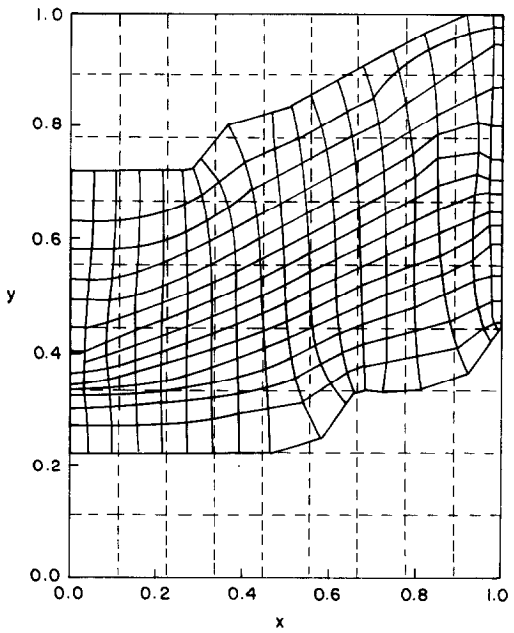


FIG. 14. Preliminary 11 x 11 coarse grid (dashed lines) and refined grid in the flagged region (solid lines) for Test Problem 3 (see additional comments in legend of Fig. 4).

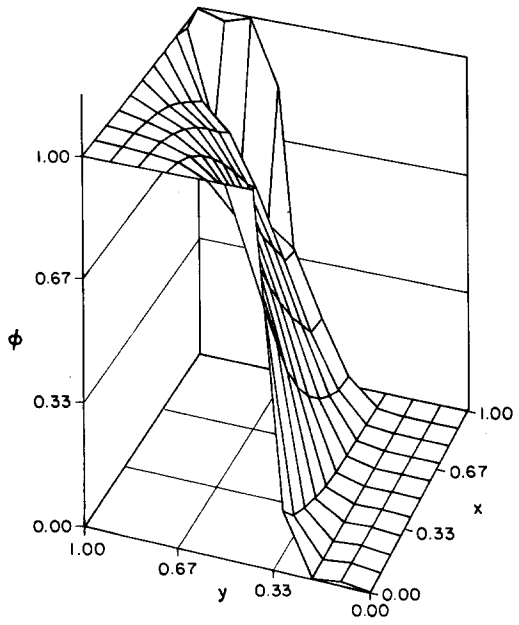


FIG. 15. The distribution of scalar variable ϕ with a conventional calculation on a 11×11 uniform grid for Test Problem 3.

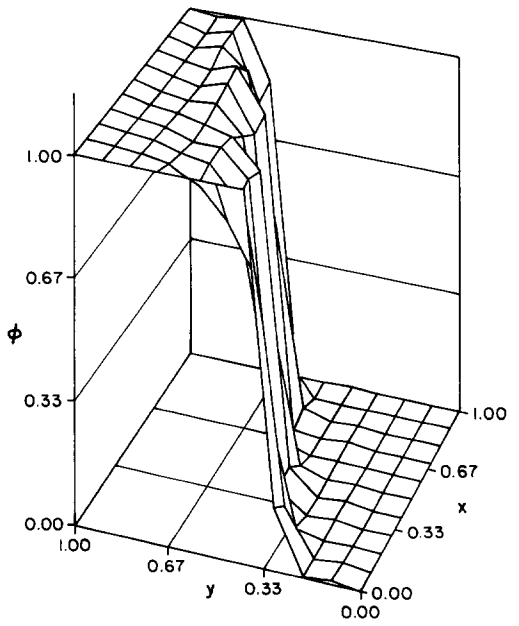


FIG. 16. The distribution of scalar variable ϕ with an adaptive grid solution procedure initiated on a 11×11 uniform grid for Test Problem 3.

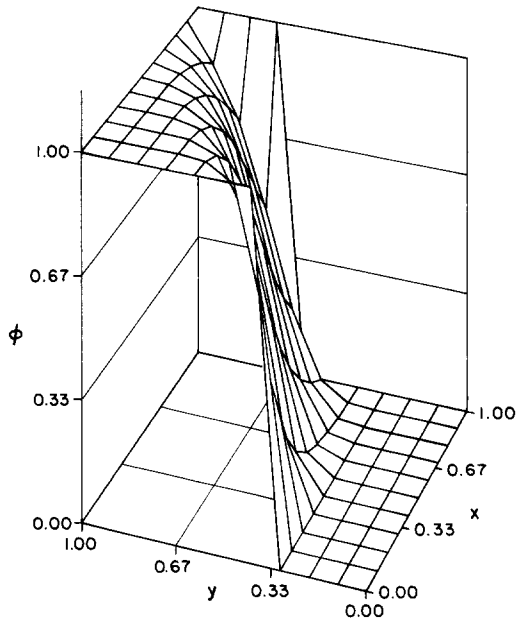


FIG. 17. The distribution of scalar variable ϕ with a conventional calculation on a 25×25 uniform grid for Test Problem 3.

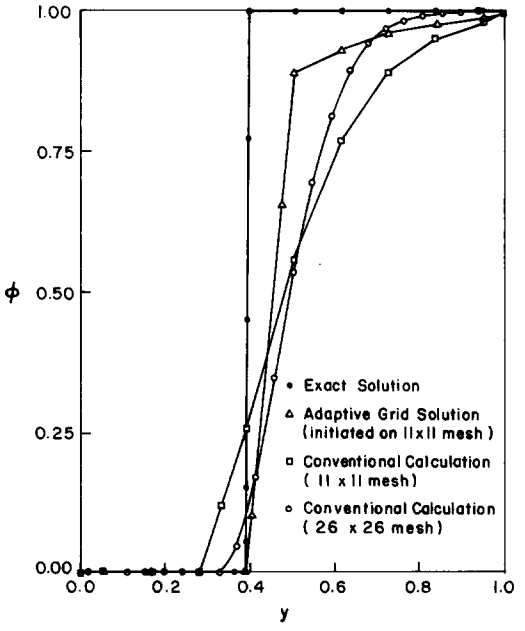


FIG. 18. Vertical centerline profile at $x = 0.44$.

in Fig. 14 which shows the original 11×11 coarse grid and the refined flagged region. As pointed out in the previous problem, to avoid staircase boundary profiles, the boundaries of the flagged region are obtained by joining with straight lines the outermost string of flagged points in the cluster. The resulting grid does appear to exhibit all the desirable features.

Figure 15 shows the ϕ profile in the $x-y$ domain obtained with a conventional calculation on a uniform 11×11 grid. A step profile is the correct solution, but the calculated profile shows a significant smearing indicative of false diffusion effects. The corresponding adaptive grid calculations initiated on the 11×11 grid are shown in Fig. 16, and the degree of smearing can be seen to be significantly reduced, and the profile is much closer to the expected step profile. The results of a conventional calculation on a uniform 25×25 grid, that requires the same effort in cpu time as the adaptive grid calculation, is shown in Fig. 17. Much greater smearing in the profile is noted, compared to the adaptive grid results. These conclusions can be best seen in Fig. 18 where the profile along the vertical centerline at $x = 0.44$ is plotted. The adaptive grid solution is seen to be superior to the conventional solution on either the 11×11 mesh or the 25×25 mesh. Thus, it might be said that the adaptive grid procedure is an effective way of minimizing false diffusion errors.

5. Concluding Remarks

An adaptive grid solution procedure is developed for convection–diffusion problems with local adaptation, multigridding, and equidistribution (LAME) concepts. The LAME procedure is initiated on a coarse grid and weighting functions or error estimates are calculated at each point from a preliminary solution on the coarse grid. Grid points are flagged if the weighting functions exceed a specified value. Thus clusters of flagged points or flagged regions are generated. The number of points in each flagged region is increased and a new mesh based on an equidistribution scheme that uses the local weighting function values is generated. The solution on the equidistributed mesh in each flagged region is calculated using boundary conditions interpolated from the coarse grid calculations. To improve the accuracy of the interpolated boundary conditions, the coarse grid solution should be improved, and to this end, a multigrid approach is adopted and coarse grid equations solved again with correction terms added to the equations in the flagged region so that the coarse grid solution in the flagged region is equal to the corresponding fine grid solution. New boundary conditions are interpolated from the improved coarse grid results and the solution in the flagged region is again obtained. This process is repeated until convergence. The method can be extended to higher levels of refinement in a multi-grid fashion.

Results are obtained for three test problems and adaptive grid results are compared with the calculations on a conventional fixed grid. The results clearly demonstrate the significant improvements obtained with the LAME procedure.

REFERENCES

1. J. F. THOMPSON, *Appl. Numer. Math.* **1**, 3 (1985).
2. S. ACHARYA AND S. V. PATANKAR, AIAA Paper No. 82-1015, 1982 (unpublished).
3. H. A. DWYER, R. J. KEE, AND B. R. SANDERS, *AIAA J.* **18**, 1025 (1980).
4. H. A. DWYER, AIAA Paper No. 83-0449, 1983 (unpublished).
5. H. A. DWYER, AIAA Paper No. 83-1932, 1983 (unpublished).
6. P. A. GNOFFO, AIAA Paper No. 82-1018, 1982 (unpublished).
7. K. NAKAHASHI AND G. S. DEIWERT, AIAA Paper No. 85-1525, 1985 (unpublished).
8. M. M. RAI AND D. A. ANDERSON, *J. Comput. Phys.* **43**, 327 (1981).
9. M. M. RAI AND D. A. ANDERSON, *AIAA J.* **20**, 496 (1982).
10. J. B. GREENBERG, AIAA Paper No. 83-1934 (unpublished).
11. P. R. EISEMAN, *J. Fluids Eng.* **107**, 477 (1985).
12. J. U. BRACKBILL AND J. SALTZMANN, *J. Comput. Phys.* **46**, 342 (1982).
13. J. U. BRACKBILL, "Coordinate System Control: Adaptive meshes," *Numerical Grid Generation*, edited by J. F. Thompson (North-Holland, Amsterdam, 1982), p. 277.
14. N. N. YANENKO, V. M. KOVENYA, V. V. LISEJKIN, V. M. FOURIN, AND E. V. VOROZHTSOV, *Comput. Methods Appl. Mech. Eng.* **17/18**, 659 (1979).
15. J. B. BELL AND G. R. SHUBIN, *J. Comput. Phys.* **52**, 569 (1983).
16. M. J. BERGER AND A. JAMESON, *AIAA J.* **23**, 561 (1985).
17. R. E. PHILLIPS AND F. W. SCHMIDT, *Numer. Heat Transfer* **8**, 573 (1985).
18. S. C. CARUSO, J. H. FERZIGER, AND J. OLIGER, AIAA Paper No. 86-0498, 1986 (unpublished).
19. J. F. THOMPSON, *Numerical Grid Generation* (North-Holland, Amsterdam, 1982).
20. F. H. MOUKALLED, Ph. D. dissertation, Louisiana State University, 1987 (unpublished).
21. D. A. ANDERSON AND J. STEINBRENNER, AIAA 24th Aerospace Science Meeting, 1986 (unpublished).
22. G. H. BRUCE, D. W. PEACEMAN, H. H. RACHFORD, AND J. D. RICE, *Trans. Amer. Inst. Min. Pet. Eng.* **198**, 79 (1953).
23. S. V. PATANKAR, *Numer. Heat Transfer* **1**, 27 (1978).
24. S. V. PATANKAR, *Numerical Heat Transfer and Fluid Flow* (Hemisphere, Washington, DC/New York, 1980).
25. P. R. EISEMAN, *Comput. Methods Appl. Mech. Eng.* **64**, 321 (1987).
26. J. P. STEINBREENER AND D. A. ANDERSON, "Three Dimensional Parametric Block Grid Regeneration with Localized Solution Adaptation," *Numerical Grid Generation in Computational Fluid Mechanics, Vol. 88*, edited by S. Sengupta, J. Hansen, P. R. Eiseman, and J. F. Thompson (Pineridge, Swansea, UK, 1988), p. 539.
27. M. A. LESCHZINER, *Comput. Methods Appl. Mech. Eng.* **23**, 293 (1980).
28. T. HAN, J. A. C. HUMPHREY, AND B. E. LAUNDER, *Comput. Methods Appl. Mech. Eng.* **29**, 81 (1981).