SIAM J. SCI. COMPUT.
Vol. 15, No. 4, pp. 776–797, July 1994

A SIMPLE ADAPTIVE GRID METHOD IN TWO DIMENSIONS*

WEIZHANG HUANG† AND DAVID M. SLOAN‡

Abstract. This paper gives an interpretation of the concept of equidistribution in the context of adaptive grid generation for multidimensional problems. It is shown that the equidistribution principle cannot be satisfied throughout the domain of the problem and, based on this recognition, a local equidistribution principle is developed. A discrete formulation is described for grid generation in two space dimensions and a smoothing mechanism is presented for improving mesh quality. The adaptive grid method that is constructed contains three grid-quality parameters. Numerical examples illustrate adaptive grid generation using a prescribed monitor function and grid generation for numerical solution of partial differential equations. Results show that the method produces high quality grids and that it is fairly insensitive to the choice of parameters.

Key words. adaptive grid, equidistribution

AMS subject classifications. 65L50, 65M50, 76M20

1. Introduction.

Numerical grid generation has become a valuable device for use in the numerical solution of partial differential equations. Many of the commonly used methods of generating computational grids are derivations of a technique first proposed by Winslow [15] in which a potential problem is solved, with mesh lines playing the role of equipotentials. Winslow's approach involves the solution of a nonlinear elliptic equation to generate a mapping from the computational domain to the physical domain. This idea was developed by J. F. Thompson and coworkers, among others, and some of these developments may be found in the text by Thompson, Warsi, and Mastin [14]. The initial developments from Winslow's method dealt with the construction of boundary-fitted coordinates for the solution of problems in irregular physical domains. The objective with these methods is to adjust the coordinate lines to match the geometry of the physical domain rather than adapt to the solution of the problem. This geometrical approach is not entirely satisfactory when a solution involves near-singular behaviour such as a boundary layer or a shock wave: in such cases the coordinate lines should be adapted to the features of the solution as well as to the geometry. Grid generators that are influenced by the solution are called adaptive grid generators.

Construction of adaptive grid generators is now an area of intense research activity, an activity that finds much of its motivation and application in computational fluid dynamics. A fairly common theme in methods used for adaptive grid generation is the idea of equidistribution, which seeks to distribute some function equally over the domain of the problem. The function is usually some measure of the computational error or the solution variation. Equidistribution has the effect of producing grids with spacing that is related to the local rate of change of the function.

The idea of equidistribution has been well defined for one-dimensional problems, but a proper description of the concept has not been produced for problems involving many dimensions. For two-dimensional problems useful progress has been made by Dwyer, Kee, and Sanders [7], Dwyer [8], and by Catherall [3]. They have used equidistribution by applying the concept in one dimension along sets of coordinate lines. In its simplest form this technique may produce grids of poor quality in terms of smoothness, skewness, and orthogonality. However, improvements in grid quality may be obtained if the equidistribution problem is

*Received by the editors August 12, 1992; accepted for publication (in revised form) April 22, 1993.

†Department of Mathematics, University of Strathclyde, Glasgow G1 1XH, United Kingdom and Institute of Applied Mathematics, Chinese Academy of Sciences, Beijing 100080, China (weizhang@cs.sfu.ca). The work of this author was supported by a Royal Fellowship Award from the Royal Society of London.

‡Department of Mathematics, University of Strathclyde, Glasgow G1 1XH, United Kingdom (caas10@ccsun.strath.ac.uk).

supplemented by conditions that deal with qualities such as smoothness and orthogonality. For example, Catherall [3] obtains good quality grids by conjoining his line-based equidistribution principle to Laplace and Poisson equations which control mesh spacings. Brackbill and Saltzman [1] used a variational approach to incorporate adaptivity (which may be regarded as a high-dimensional extension of the one-dimensional equidistribution principle) into Winslow's [15] method. They included terms in their variational formulation designed to improve grid properties such as smoothness and orthogonality. Dorfi and Drury [5], in considering one-dimensional problems, used a very effective device for incorporating smoothness into the equidistribution principle. Their one-dimensional device ensures that the ratio of adjacent grid intervals is restricted, thus controlling clustering and grid expansion. The power of the smoothing capability in [5] is clearly demonstrated in the valuable comparative study by Furzland, Verwer, and Zegeling [9].

The objective of this paper is to clarify the concept of equidistribution for multidimensional problems and to describe a simple method which produces high-quality meshes in two dimensions. The equidistribution principle is described in continuous form in §2.1 and in discrete form in §2.2. A mechanism for improving mesh quality, based on the ideas of Dorfi and Drury [5], is described in §2.3, and a simple 3-parameter generator is presented. Adaptive grid generation using a prescribed function is described in §3, and generation associated with the numerical solution of differential equations in two spatial dimensions is described in §4. Numerical examples are presented in §§3 and 4. Conclusions and comments on our 3-parameter method are contained in §5.

2. Simple grid adaption based on equidistribution principles.

2.1. Equidistribution principles. The equidistribution idea, first used by de Boor in 1974 [4], is the most important concept in the field of adaptive grid generation. Most adaptive grid methods are related in some way to this concept (see, for example, Hawken, Gottlieb, and Hansen [10]). However, although equidistribution is understood in one space dimension, it does not seem to have been well formulated mathematically in more than one dimension.

In this subsection we shall construct equidistribution principles based on the idea of equivariation of certain functions. Let $\mathbf{x} = [x_1, x_2, x_3]^T$ be the spatial coordinate in a three-dimensional physical domain, D_p, where the superscript T denotes transposition. Suppose we select a function u on D_p whose variation is being considered. To simplify the presentation, we introduce a one-to-one coordinate transformation from the computational domain D_c to the physical domain D_p as

$$\mathbf{x} = \mathbf{x}(\boldsymbol{\xi}), \qquad \boldsymbol{\xi} \in D_c, \tag{1}$$

where $\boldsymbol{\xi} = [\xi^1, \xi^2, \xi^3]^T$ denotes the spatial coordinate on the computational domain D_c. Obviously, for a prescribed mesh on the computational domain, the problem of finding a corresponding mesh on the physical domain will be that of determining a discrete transformation (1), which is defined for all $\boldsymbol{\xi}$ on the given mesh. Henceforth, the determination of a mesh, or a distribution of nodes, on D_p will be regarded as being equivalent to the construction of a (discrete) transformation (1). A distribution of the nodes of a mesh will hereinafter be referred to as a distribution.

It is well known (see, for example, Thompson, Warsi, and Mastin [14]) that the scaled arc-length measurement of variation of u along the element of arc from $\mathbf{x}$ to $\mathbf{x} + d\mathbf{x}$ can be expressed as

$$ds = [\alpha^2 (du)^2 + d\mathbf{x}^T\, d\mathbf{x}]^{1/2} = [d\mathbf{x}^T\, M\, d\mathbf{x}]^{1/2}, \tag{2}$$

where M is defined by

$$M = \alpha^2 \nabla u \cdot \nabla u^T + I, \tag{3}$$

$$\nabla u = \left[\frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2}, \frac{\partial u}{\partial x_3} \right]^T, \tag{4}$$

and I is the unit matrix. An appreciation of scaled arc-length is more readily obtained by considering the one-dimensional situation, with x and ξ in D_p and D_c, respectively. An increment of arc-length, ds, on the solution curve $u = u(x)$ between locations x and $x + dx$ in D_p is given by

$$ds = [(du)^2 + (dx)^2]^{1/2} = \left[\left(\frac{du}{dx} \right)^2 + 1 \right]^{1/2} dx.$$

To obtain a scaled increment, which allows the relative effects of changes in u and x to be altered, we may replace the above ds by

$$ds = \left[\alpha^2 \left(\frac{du}{dx} \right)^2 + 1 \right]^{1/2} dx,$$

where $\alpha^2 \geq 0$ is a real parameter. If $\alpha^2 = 0$, the increment simply measures the change in x, and as α^2 increases the influence of the variation in u becomes more significant. Returning now to the multidimensional problem, if we use transformation (1), equation (2) can be expressed in computational coordinates as

$$ds = \left[d\boldsymbol{\xi}^T J^T M J d\boldsymbol{\xi} \right]^{1/2}, \tag{5}$$

where

$$J = \left[\frac{\partial \mathbf{x}}{\partial \xi^1}, \frac{\partial \mathbf{x}}{\partial \xi^2}, \frac{\partial \mathbf{x}}{\partial \xi^3} \right] \tag{6}$$

is the Jacobian of coordinate transformation (1).

The equidistribution principle follows from a simple observation on (5): if $u(\mathbf{x}(\boldsymbol{\xi}))$ is required to have the same variation ds along any element of arc in the computational domain which has fixed length $[d\boldsymbol{\xi}^T d\boldsymbol{\xi}]^{1/2}$, the term on the right-hand side of (5) must be independent of coordinate $\boldsymbol{\xi}$. Therefore, this equivariation requirement on u implies that $J^T M J$ should be independent of $\boldsymbol{\xi}$; that is,

$$[d\boldsymbol{\xi}^T J^T M J \, d\boldsymbol{\xi}]^{1/2} = [d\boldsymbol{\xi}^T \bar{M} d\boldsymbol{\xi}]^{1/2}, \tag{7}$$

where $\bar{M}$ is a 3×3 symmetric positive definite, and $\boldsymbol{\xi}$-independent, matrix. Thus, if a coordinate transformation (1) can be found to satisfy (7), u will have the same variation at any point on D_p along any arc that has length

$$\left[\left(\sum_{i=1}^{3} \frac{\partial \mathbf{x}}{\partial \xi^i} d\xi^i \right)^T \left(\sum_{j=1}^{3} \frac{\partial \mathbf{x}}{\partial \xi^j} d\xi^j \right) \right]^{1/2}.$$

In this sense, any one-to-one coordinate transformation (1) that satisfies (7) for some constant, symmetric positive definite matrix $\bar{M}$ will be called an *equidistribution*, and (7) will be called an *equidistribution principle* (in continuous form).

Now let's consider how we might use the equidistribution principle (7) to determine a distribution on D_p (physical mesh) corresponding to a given uniform mesh on D_c (computational mesh). The equidistribution principle (7) may be discretised by any suitable discretisation method, and this process yields an equation corresponding to each link between adjacent pairs of nodes on the computational mesh. The physical mesh may then be obtained by solving these equations, in conjunction with some discrete boundary conditions for the coordinate transformation (1). Some method has to be given for the determination of the constant matrix $\bar{M}$ in (7). In the following, we shall consider two common cases.

Case 1. The number of nodes is not restricted. This case often occurs in unstructured mesh generation and refinement mesh generation. Noting that M defined by (3) is a symmetric and positive definite matrix, we may choose the coordinate transformation (1) such that $\bar{M}$ has the special form

$$\bar{M} = c^2 I, \tag{8}$$

where $c > 0$ is a constant. For example, this can be achieved by choosing the computational coordinate system to coincide with the principal directions of $\bar{M}$. Therefore, (7) takes the form

$$\left[d\xi^T J^T M J d\xi\right]^{1/2} = c\left[d\xi^T d\xi\right]^{1/2}. \tag{9}$$

In this case, c may be given in advance. Similar ideas have been used for the adaptive generation of unstructured meshes (see, for example, Peraire et al. [13], Bristeau et al. [2]).

Case 2. The number of nodes along any coordinate line is given in advance and the global structure of the mesh is also required. This is the case in structured movement (or redistributed) mesh generation. Usually (7) cannot be satisfied by the coordinate transformation (1) (or by a mesh on D_p) on the whole computational domain. In fact, this can be seen easily from the following analysis.

Suppose (7) can be satisfied by some coordinate transformation on the whole computational domain. Then (7) must hold along any coordinate line. For example, along the coordinate line

$$l_1 : \xi^2 = \text{constant}, \qquad \xi^3 = \text{constant}, \tag{10}$$

(7) reads as

$$\left[\left(\frac{\partial \mathbf{x}}{\partial \xi^1}\right)^T M \left(\frac{\partial \mathbf{x}}{\partial \xi^1}\right)\right]^{1/2} = c_1, \tag{11}$$

where c_1 is a positive constant. On the other hand, the equidistribution along line l_1 can be regarded as a one-dimensional case. Therefore, the assumption that the number of nodes along l_1 is prescribed implies that c_1 can be determined by the equidistribution, and this gives

$$c_1 = \frac{\displaystyle\int \left[\left(\frac{\partial \mathbf{x}}{\partial \xi^1}\right)^T M \left(\frac{\partial \mathbf{x}}{\partial \xi^1}\right)\right]^{1/2} d\xi^1}{\displaystyle\int d\xi^1}, \tag{12}$$

where the integration is along l_1 in D_c. If the number of nodes on l_1 is prescribed in advance, then the denominator on the right-hand side of (12) is fixed at a value determined by this number. Hence, when the number of nodes along any coordinate line is given in advance, it may be seen that c_1 will usually be a function of ξ^2 and ξ^3. Since c_1 in (11) is assumed to be constant over the entire computational domain it follows that there does not exist an equidistribution on D_p for this case. However, if we weaken (7) and require the coordinate transformation (1) only to satisfy (7) locally, it will be possible to find an equidistribution on D_p. Of course, this equidistribution is only *locally-equidistributed.* Here we consider (7) to be satisfied along any coordinate line: for example, along line l_1 we have (11), but with c_1 now a function of ξ^2 and ξ^3. Similarly, along the other two coordinate lines analogous formulae can be obtained. Hence, we have

$$\tag{13} \begin{cases} \left[\left(\frac{\partial \mathbf{x}}{\partial \xi^1}\right)^T M \left(\frac{\partial \mathbf{x}}{\partial \xi^1}\right)\right]^{1/2} = c_1(\xi^2, \xi^3), \\ \left[\left(\frac{\partial \mathbf{x}}{\partial \xi^2}\right)^T M \left(\frac{\partial \mathbf{x}}{\partial \xi^2}\right)\right]^{1/2} = c_2(\xi^3, \xi^1), \\ \left[\left(\frac{\partial \mathbf{x}}{\partial \xi^3}\right)^T M \left(\frac{\partial \mathbf{x}}{\partial \xi^3}\right)\right]^{1/2} = c_3(\xi^1, \xi^2). \end{cases}$$

The equations in (13) give the essential formulae in our simple adaptive grid method. The idea of using a curve-based equidistribution principle is not new, having been used by Dwyer, Kee, and Sanders [7], Dwyer [8], and Catherall [3]. However, it does not seem to have been presented in a form as simple as that offered by (13).

It is instructive to look at the two- and one-dimensional forms of (13).

Two Dimensions

$$\tag{14} \begin{cases} \left[\begin{pmatrix} \frac{\partial x}{\partial \xi} \\ \frac{\partial y}{\partial \xi} \end{pmatrix}^T M \begin{pmatrix} \frac{\partial x}{\partial \xi} \\ \frac{\partial y}{\partial \xi} \end{pmatrix}\right]^{1/2} = c_1(\eta), \\ \left[\begin{pmatrix} \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \eta} \end{pmatrix}^T M \begin{pmatrix} \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \eta} \end{pmatrix}\right]^{1/2} = c_2(\xi), \end{cases}$$

where

$$\tag{15} M = \alpha^2 \begin{bmatrix} \left(\frac{\partial u}{\partial x}\right)^2 & \frac{\partial u}{\partial x}\frac{\partial u}{\partial y} \\ \frac{\partial u}{\partial x}\frac{\partial u}{\partial y} & \left(\frac{\partial u}{\partial y}\right)^2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

and (ξ, η) denotes the spatial coordinate in D_c.

One Dimension

$$\tag{16} \left[\left(\frac{dx}{d\xi}\right)^2 M\right]^{1/2} = c,$$

where

$$\tag{17} M = 1 + \alpha^2 \left(\frac{du}{dx}\right)^2,$$

and ξ is the spatial coordinate in D_c. Equation (16) is the well-known, one-dimensional arc-length equidistribution principle.

The remainder of this paper deals with structured, redistributed, adaptive mesh generation in two dimensions.

2.2. Finite difference discretisation of the two-dimensional equidistribution principle. In this subsection, a finite difference discretisation of (14) will be presented. Suppose that a uniform mesh is given on D_c by

$$\left\{\begin{array}{ll} \xi_i = i, & i = 0, \ldots, n, \\ \eta_j = j, & j = 0, \ldots, m, \end{array}\right. \tag{18}$$

where n and m denote the numbers of intervals in the ξ and η directions, respectively. Equations (14) are discretized on the internodal links containing $(i + \frac{1}{2}, j)$ and $(i, j + \frac{1}{2})$, respectively, to give

$$\begin{gathered}\left\{\begin{bmatrix} x_{i+1,j}-x_{i,j} \\ y_{i+1,j}-y_{i,j} \end{bmatrix}^T M_{i+\frac{1}{2},j} \begin{bmatrix} x_{i+1,j}-x_{i,j} \\ y_{i+1,j}-y_{i,j} \end{bmatrix}\right\}^{1/2} = c_1(\eta_j), \\ i = 0, \ldots, n-1, \qquad j = 1, \ldots, m-1, \end{gathered} \tag{19}$$

and

$$\begin{gathered}\left\{\begin{bmatrix} x_{i,j+1}-x_{i,j} \\ y_{i,j+1}-y_{i,j} \end{bmatrix}^T M_{i,j+\frac{1}{2}} \begin{bmatrix} x_{i,j+1}-x_{i,j} \\ y_{i,j+1}-y_{i,j} \end{bmatrix}\right\}^{1/2} = c_2(\xi_i), \\ j = 0, \ldots, m-1, \qquad i = 1, \ldots, n-1, \end{gathered} \tag{20}$$

where $M_{i+1/2,j} = M(\xi_{i+1/2}, \eta_j)$ and $M_{i,j+1/2} = M(\xi_i, \eta_{j+1/2})$. Equations (19) and (20) may be rewritten as

$$\begin{gathered}\left\{\begin{bmatrix} x_{i,j}-x_{i-1,j} \\ y_{i,j}-y_{i-1,j} \end{bmatrix}^T M_{i-\frac{1}{2},j} \begin{bmatrix} x_{i,j}-x_{i-1,j} \\ y_{i,j}-y_{i-1,j} \end{bmatrix}\right\}^{1/2} \\ - \left\{\begin{bmatrix} x_{i+1,j}-x_{i,j} \\ y_{i+1,j}-y_{i,j} \end{bmatrix}^T M_{i+\frac{1}{2},j} \begin{bmatrix} x_{i+1,j}-x_{i,j} \\ y_{i+1,j}-y_{i,j} \end{bmatrix}\right\}^{1/2} = 0, \\ i = 1, \ldots, n-1, \qquad j = 1, \ldots, m-1, \end{gathered} \tag{21}$$

$$\begin{gathered}\left\{\begin{bmatrix} x_{i,j}-x_{i,j-1} \\ y_{i,j}-y_{i,j-1} \end{bmatrix}^T M_{i,j-\frac{1}{2}} \begin{bmatrix} x_{i,j}-x_{i,j-1} \\ y_{i,j}-y_{i,j-1} \end{bmatrix}\right\}^{1/2} \\ - \left\{\begin{bmatrix} x_{i,j+1}-x_{i,j} \\ y_{i,j+1}-y_{i,j} \end{bmatrix}^T M_{i,j+\frac{1}{2}} \begin{bmatrix} x_{i,j+1}-x_{i,j} \\ y_{i,j+1}-y_{i,j} \end{bmatrix}\right\}^{1/2} = 0, \\ j = 1, \ldots, m-1, \qquad i = 1, \ldots, n-1. \end{gathered} \tag{22}$$

Regarding the discretisation of matrix M, it needs to be transformed into computational coordinates and then discretised on the computational domain. To be more specific, M is transformed into

$$M(\xi, \eta) = \alpha^2 \begin{bmatrix} \frac{u_\xi y_\eta - u_\eta y_\xi}{\sqrt{g}} \\ \frac{-u_\xi x_\eta + u_\eta x_\xi}{\sqrt{g}} \end{bmatrix} \cdot \left[\frac{u_\xi y_\eta - u_\eta y_\xi}{\sqrt{g}}, \frac{-u_\xi x_\eta + u_\eta x_\xi}{\sqrt{g}}\right] + I, \tag{23}$$

where $\sqrt{g} = x_\xi y_\eta - x_\eta y_\xi$. $M(\xi, \eta)$ can now be discretised at point $(\xi_{i+1/2}, \eta_{j+1/2})$ by second-order central differences. For example, in M, u_ξ and u_η can be discretised by

$$u_\xi\left(\xi_{i+\frac{1}{2}}, \eta_{j+\frac{1}{2}}\right) = \frac{1}{2}\left(u_{i+1,j} - u_{i,j} + u_{i+1,j+1} - u_{i,j+1}\right)$$

and

$$u_\eta\left(\xi_{i+\frac{1}{2}}, \eta_{j+\frac{1}{2}}\right) = \frac{1}{2}\left(u_{i,j+1} - u_{i,j} + u_{i+1,j+1} - u_{i+1,j}\right),$$

respectively, where $u_{i,j} = u(x_{i,j}, y_{i,j}) = u\big(x(\xi_i, \eta_j), y(\xi_i, \eta_j)\big)$.

Then $M_{i+1/2,j}$ and $M_{i,j+1/2}$ can be obtained by

$$\text{(24)}\qquad \begin{aligned} M_{i+\frac{1}{2},j} &= \frac{1}{2}\left(M_{i+\frac{1}{2},j+\frac{1}{2}} + M_{i+\frac{1}{2},j-\frac{1}{2}}\right), \\ M_{i,j+\frac{1}{2}} &= \frac{1}{2}\left(M_{i+\frac{1}{2},j+\frac{1}{2}} + M_{i-\frac{1}{2},j+\frac{1}{2}}\right). \end{aligned}$$

2.3. Smoothness of mesh and the simple grid adaption. It is known that the mesh quality, such as smoothness of mesh and orthogonality along boundaries, is another important aspect of grid adaption (see, for example, Thompson, Warsi, and Mastin [14]). As seen in previous subsections, the equidistribution principle does not offer any mechanism to handle the mesh quality. Some mechanism should therefore be incorporated into the equidistribution principle in order to control the quality. In the popular variational approach to grid adaption, the balance between grid adaptivity and mesh quality is introduced by using a linear combination of terms associated with adaptivity, smoothness, and orthogonality. Recently, the same idea was used by Catherall [3]. He combines the equidistribution principle linearly with a Laplace and a Poisson equation for the nodal locations, with the combination designed to improve the grid quality. Suitable choices of the balance parameters can give good quality meshes. However, the proper balance depends on the physical problem and the ideal selection strategy is still an open question. An insensitive way to smooth the mesh in one dimension has been used by Dorfi and Drury [5] (see also Furzland, Verwer, and Zegeling [9]). They achieve their objective by smoothing the matrix M given by (17).

Here we follow the approach proposed by Dorfi and Drury [5]. In this approach, matrices (21) and (22) are replaced by smoothing matrices

$$\text{(25)}\qquad \begin{aligned} \tilde{M}_{i+\frac{1}{2},j} &= \sum_{k=i-1}^{i+1} \sum_{l=j-1}^{j+1} M_{k+\frac{1}{2},l} \left(\frac{\gamma}{\gamma+1}\right)^{|k-i|+|l-j|}, \\ \tilde{M}_{i,j+\frac{1}{2}} &= \sum_{k=i-1}^{i+1} \sum_{l=j-1}^{j+1} M_{k,l+\frac{1}{2}} \left(\frac{\gamma}{\gamma+1}\right)^{|k-i|+|l-j|}, \end{aligned}$$

where γ, called the smoothing parameter, is a positive constant. The summations in (25) are understood to contain only elements that are well defined. It is of interest to note that the smoothing introduced by (25) couples three coordinate lines in each direction ($l = j - 1, j, j + 1$ in the first equation of (25), for example). The smoothing process adds numerical dissipation, rather like an increase in viscosity in a fluid flow—a process that reduces the tendency for fluid shear. The analogy here is that the smoothing produces an effect that prevents the mesh from skewing. Using (25), equations (21), and (22) read as

$$(26)\quad \left\{\begin{array}{l} \left\{\begin{bmatrix} x_{i,j}-x_{i-1,j} \\ y_{i,j}-y_{i-1,j} \end{bmatrix}^T \tilde{M}_{i-\frac{1}{2},j} \begin{bmatrix} x_{i,j}-x_{i-1,j} \\ y_{i,j}-y_{i-1,j} \end{bmatrix}\right\}^{1/2} \\ \qquad -\left\{\begin{bmatrix} x_{i+1,j}-x_{i,j} \\ y_{i+1,j}-y_{i,j} \end{bmatrix}^T \tilde{M}_{i+\frac{1}{2},j} \begin{bmatrix} x_{i+1,j}-x_{i,j} \\ y_{i+1,j}-y_{i,j} \end{bmatrix}\right\}^{1/2} = 0, \\ \left\{\begin{bmatrix} x_{i,j}-x_{i,j-1} \\ y_{i,j}-y_{i,j-1} \end{bmatrix}^T \tilde{M}_{i,j-\frac{1}{2}} \begin{bmatrix} x_{i,j}-x_{i,j-1} \\ y_{i,j}-y_{i,j-1} \end{bmatrix}\right\}^{1/2} \\ \qquad -\left\{\begin{bmatrix} x_{i,j+1}-x_{i,j} \\ y_{i,j+1}-y_{i,j} \end{bmatrix}^T \tilde{M}_{i,j+\frac{1}{2}} \begin{bmatrix} x_{i,j+1}-x_{i,j} \\ y_{i,j+1}-y_{i,j} \end{bmatrix}\right\}^{1/2} = 0, \\ \qquad\qquad\qquad\qquad i = 1, \dots, n-1, \\ \qquad\qquad\qquad\qquad j = 1, \dots, m-1. \end{array}\right.$$

The adaptive grid method presented here consists of (3), (25), (26), and suitable boundary conditions at $\xi = \xi_0, \xi_n$ and $\eta = \eta_0, \eta_m$. It should be noted that low-order discretisations are often used along the boundaries, and these should take the form of mesh orthogonality conditions.

In practical calculations involving very stiff problems it is found that dangerously small physical grid spacings and mesh tangling ("overkill") can occur when $|\nabla u|$ becomes too large somewhere. This problem can be avoided by imposing a cut-off condition on ∇u in (3). That is, (3) is replaced by

$$(27)\qquad M = \frac{\alpha^2 \nabla u \,.\, \nabla u^T}{1 + \beta \nabla u^T \nabla u} + I,$$

where β, called a scaling parameter, is a nonnegative constant. Equation (27) implies that

$$(28)\qquad \|M\|_2 \le 1 + \frac{\alpha^2}{\beta}$$

for positive β. Therefore, positive β imposes minimum values on the increments of x and y along every computational coordinate line and thus obviates "overkill."

Thus, the three parameters α (concentration), β (scaling), and γ (smoothing) are introduced in the current adaptive grid method. The sensitivity of choice of these parameters will be discussed in the following sections.

3. Grid generation for given analytic functions. Before we apply the adaptive grid method to numerical solutions of differential equations, it will be useful to look at how to choose values of the parameters and how good meshes can be produced for some given function u. Two functions defined on a rectangular physical domain will be considered in this section. Dirichlet and Neumann (orthogonal) boundary conditions for x and y are used, that is,

$$(29)\qquad \left\{\begin{array}{ll} \left\{\begin{array}{l} x_{0,j} = 0, \;\; x_{n,j} = 1, \\ y_{1,j} - y_{0,j} = 0, \;\; y_{n,j} - y_{n-1,j} = 0, \end{array}\right. & j = 0, \dots, m, \\ \left\{\begin{array}{l} x_{i,1} - x_{i,0} = 0, \;\; x_{i,m} - x_{i,m-1} = 0, \\ y_{i,0} = 0, \;\; y_{i,m} = 1. \end{array}\right. & i = 1, \dots, n-1, \end{array}\right.$$

The nonlinear system (29) and (26) is solved by homotopy and a Newton–Raphson method, wherein α serves as continuation parameter ($0 \rightarrow \alpha$). The Jacobian is calculated by forward differences and linear systems arising from Newton linearisation are solved by Gaussian elimination. The uniform mesh is used as the initial mesh.

Example 3.1. In this example we use the analytic function

$$u(x,y) = [1 - e^{R(x-1)}]\sin(\omega y). \tag{30}$$

This function is the exact solution of the convection-diffusion equation which is considered in §4 (Example 4.1) and it has been used for grid generation by Dvinsky [6]. When R is large, $u(x,y)$ has a boundary layer near $x = 1$. In this example, the scaling effect has not been considered ($\beta = 0$).

First, let us consider the case $R = 15$ and $\omega = \pi$. The exact function (on physical domain), meshes, and functions evaluated on these meshes are shown in Fig. 3.1. From Fig. 3.1 we make some observations:

a. Adapted meshes can be obtained by the equidistribution principle. However, meshes may be very unsmooth and skewed if $\gamma = 0$ (without control of smoothness).
b. The method described in §2.2 offers a good mechanism for handling mesh smoothness. The choice of values of γ is not sensitive.
c. The larger the values of α, the more adaptive the mesh becomes. The choice of values of α depends on how much adaptivity is needed. Usually, $\alpha = 1$ will suffice.
d. The more grid points we use, the better the quality of the generated mesh.

Adapted meshes and functions on these meshes are shown in Figs. 3.2 and 3.3 for the case $R = 35, \omega = \pi$, and the case $R = 15, \omega = 1.5\pi$, respectively. These figures confirm our observations.

Note that with Neumann boundary conditions, orthogonality of the grid lines is imposed at the boundary, and this may lead to discontinuities in grid line slope at points adjacent to the boundary. This did not give rise to noticeable inaccuracies for the problems considered here since the resulting skewness at points adjacent to the boundaries was never too severe. The slope discontinuities could be avoided if the Neumann conditions were replaced by an imposition of the equidistribution principle on the boundary lines.

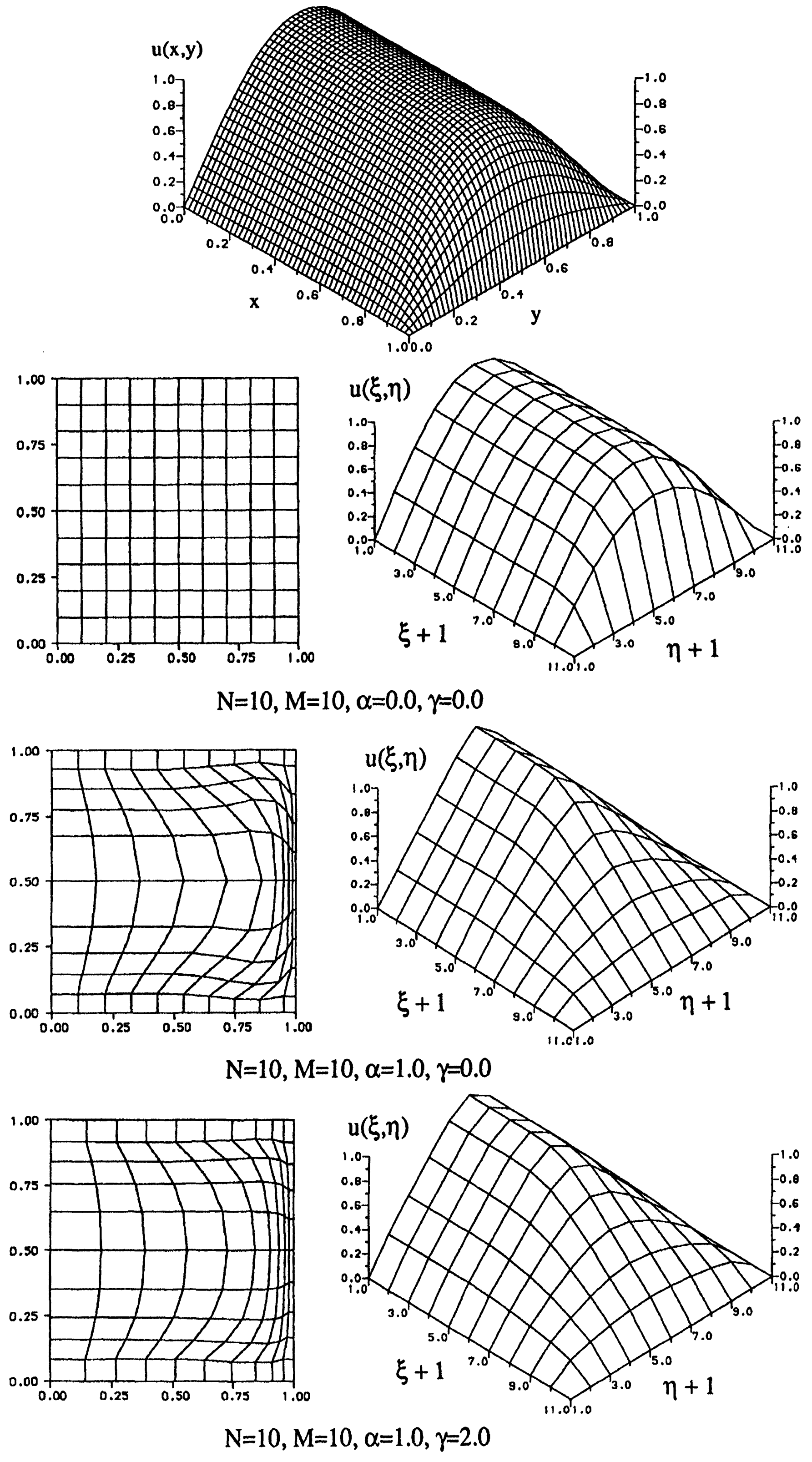
Example 3.2. Here we use the analytic function

$$u(x,y) = \tanh\left[R\left(\frac{1}{16} - \left(x - \frac{1}{2}\right)^2 - \left(y - \frac{1}{2}\right)^2\right)\right]. \tag{31}$$

This is a more difficult problem than Example 3.1 because the mesh needs to be adapted in all directions. The function, with several values of R, is plotted in Fig. 3.4.

Adapted meshes obtained by the adaptive grid method described in §2 with $\alpha = 1, \gamma = 2$, and $\beta = 0$ (no scaling) are shown in Fig. 3.5. Although a good mesh can be obtained for $R = 15$, the "overkill" phenomenon appears for larger $R(= 20)$ near the four corners, where $\nabla u^T \nabla u$ is very large. As suggested in §2.2, the "overkill" phenomenon may be avoided by cutting off (or scaling) ∇u. Figures 3.6 and 3.7 show adapted meshes obtained with $\beta = 0.01$ (five times smaller than $\frac{1}{n} = \frac{1}{m} = 0.05$) and with $\beta = 0.005$ (ten times smaller than $\frac{1}{n} = \frac{1}{m} = 0.05$), respectively. It can be seen that the "overkill" phenomenon disappears even for much larger $R(= 100)$ and good quality meshes can be obtained for a wide range of values of β. It is also interesting to notice from Figs. 3.5–3.7 that positive β provides a safe guarantee for large values of R.

In summary, it has been shown that adapted meshes with good quality can be obtained for given analytic monitor functions by using the simple adaptive grid method described in §2. The ideal values of the parameters α, β, and γ will depend on the monitor function. The method is fairly insensitive to the choice of values of the parameters, however, and we

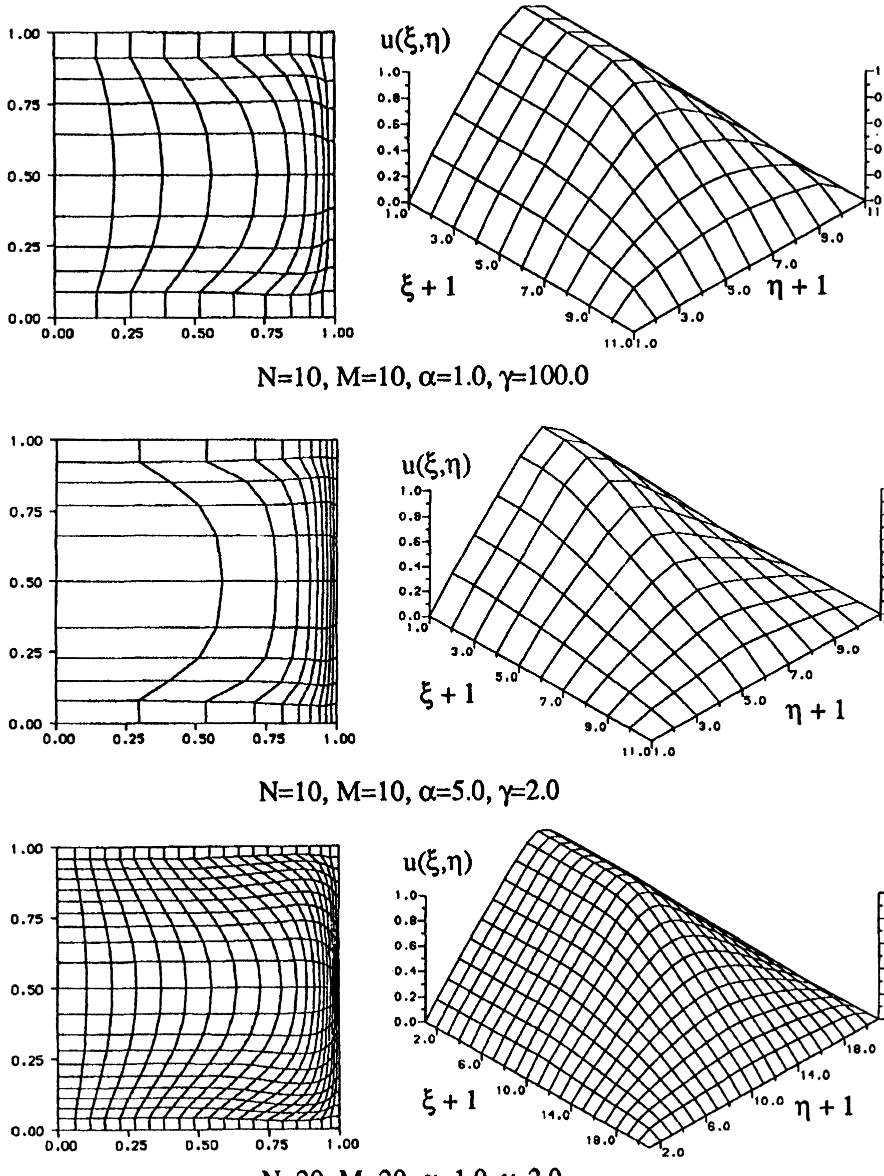


N=10, M=10, α=0.0, γ=0.0

N=10, M=10, α=1.0, γ=0.0

N=10, M=10, α=1.0, γ=2.0

FIG. 3.1. *Solution and meshes for Example* 3.1 *with* $R = 15.0$, $\omega = \pi$.



N=10, M=10, α=1.0, γ=100.0

N=10, M=10, α=5.0, γ=2.0

N=20, M=20, α=1.0, γ=2.0

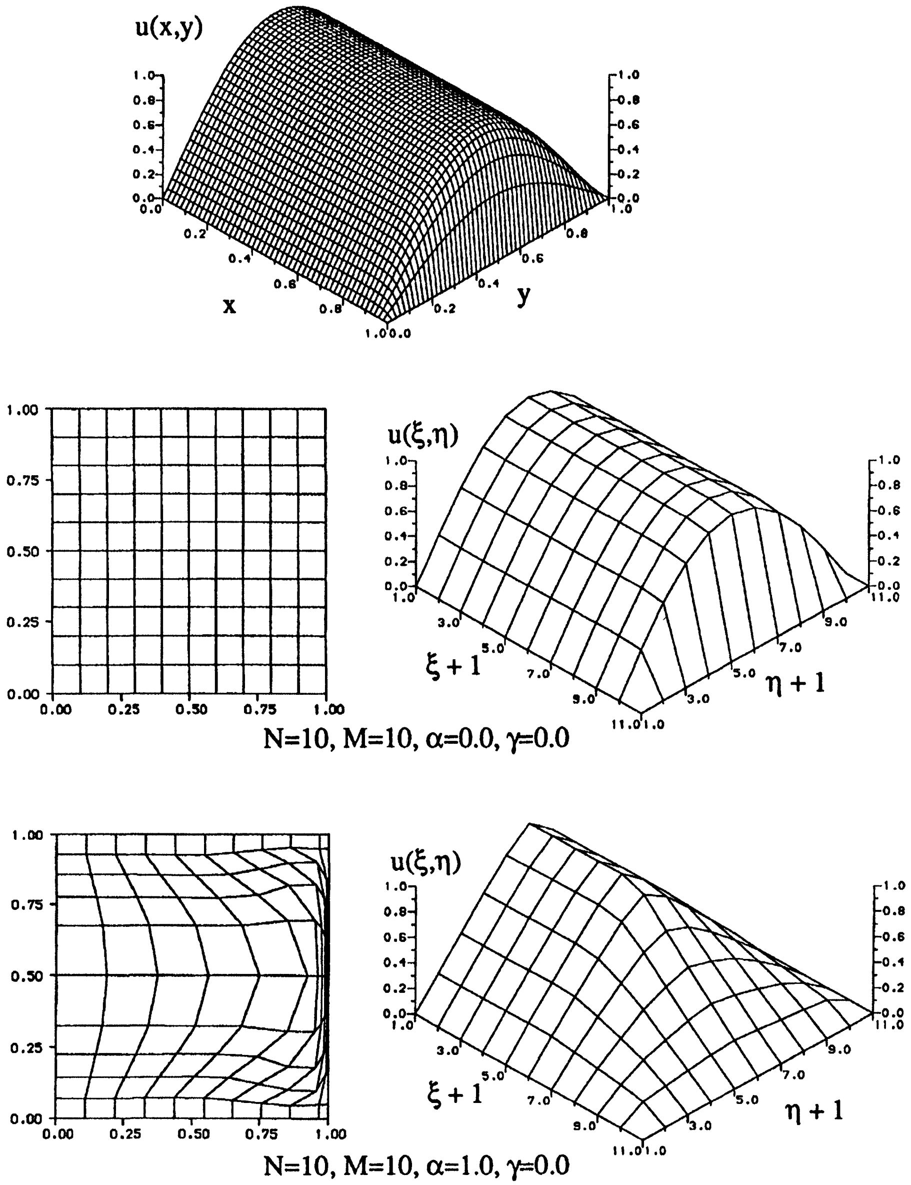
FIG. 3.1. *Continued.*

experienced no trouble in selecting suitable values, even in the difficult case of Example 3.2 with $R \geq 20$.

4. Grid adaption for numerical solutions of differential equations in two dimensions. In this section, the adaptive grid method described in §2 is applied to numerical solutions of the convection-diffusion equation given below.

Example 4.1. Consider

$$R\frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \omega^2\left[1 - e^{R(x-1)}\right]\sin(\omega y), \qquad 0 < x, y < 1, \tag{32}$$



N=10, M=10, α=0.0, γ=0.0

N=10, M=10, α=1.0, γ=0.0

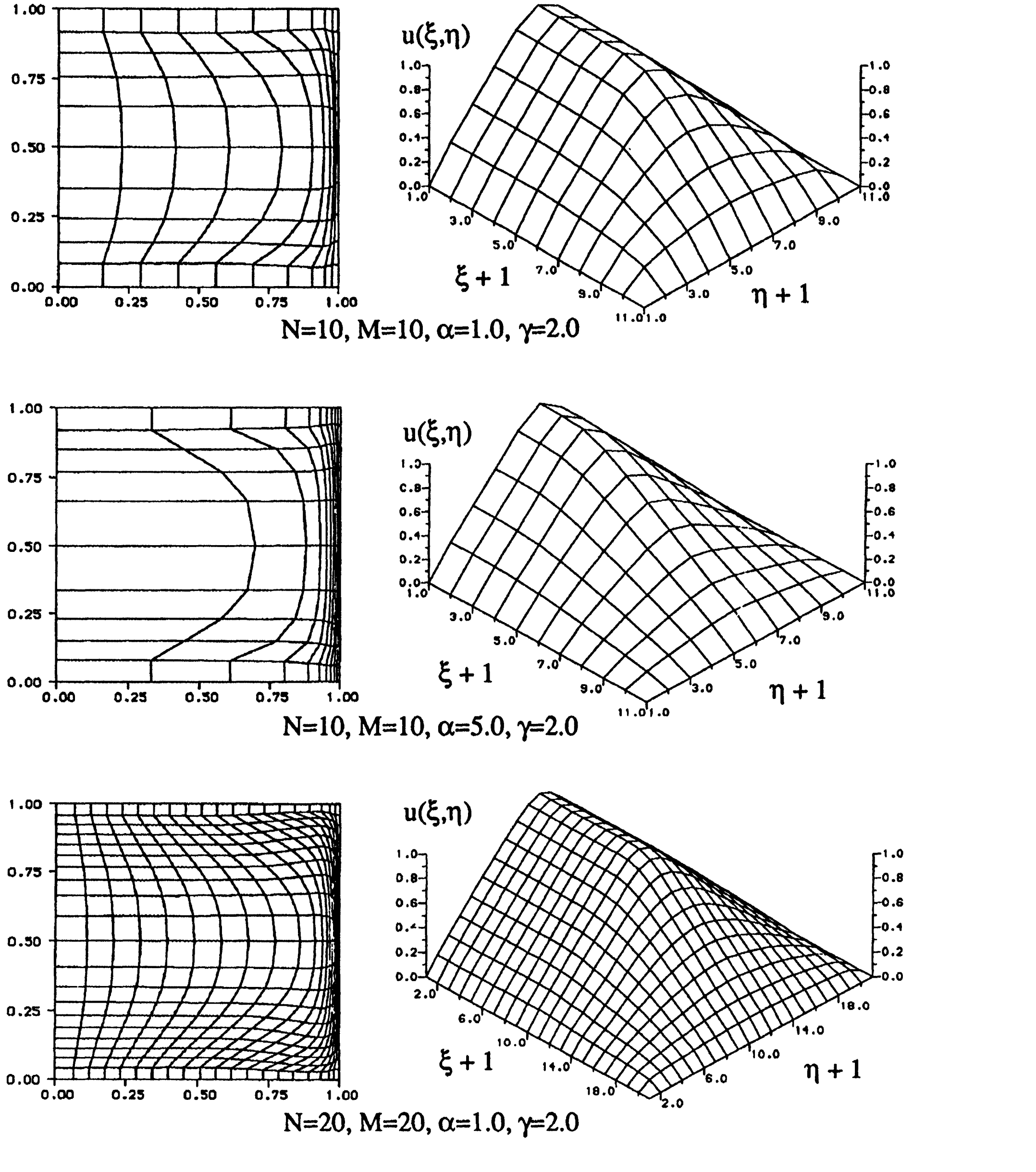
FIG. 3.2. *Solution and meshes for Example* 3.1 *with* $R = 35.0$, $\omega = \pi$.

subject to Dirichlet boundary conditions that are chosen such that (32) has the exact solution

$$u^e(x, y) = \left[1 - e^{R(x-1)}\right] \sin(\omega y). \tag{33}$$

This differential equation has been used by Dvinsky [6] to show the efficiency of his adaptive grid method.

In our calculations, (32) is first transformed into computational coordinates and then discretised on the uniform mesh (18) by first-order upwind differences. The resulting algebraic equations will hereinafter be called discrete convection-diffusion equations (DCDE) for simplicity of description. The equations (26), (29), and DCDE constitute the algebraic system



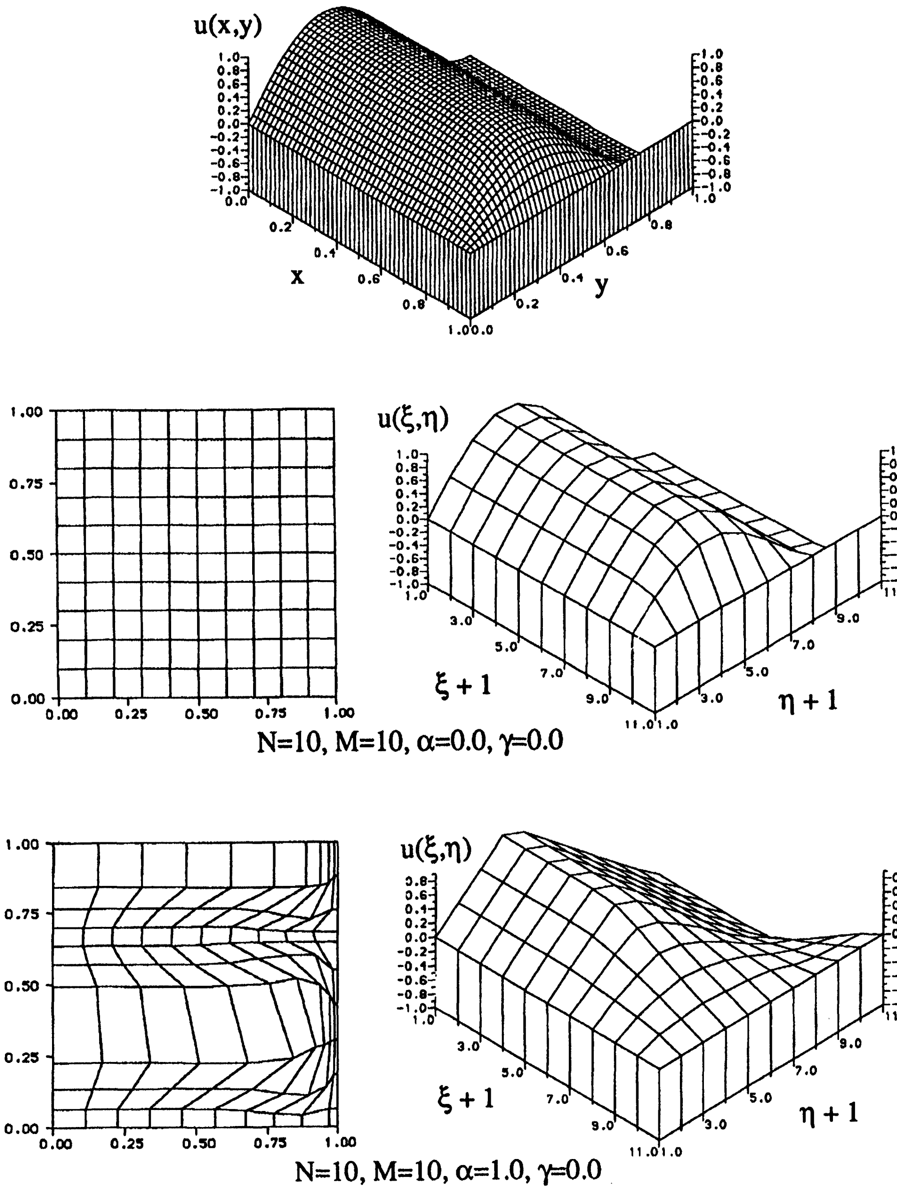
N=10, M=10, α=1.0, γ=2.0

N=10, M=10, α=5.0, γ=2.0

N=20, M=20, α=1.0, γ=2.0

FIG. 3.2. *Continued.*

that is needed to determine the unknowns $u_{i,j}, x_{i,j}$, and $y_{i,j}$, $i = 0, \dots, n$, $j = 0, 1, \dots, m$. Obviously, the equations in this algebraic system should be solved simultaneously. However, the complete system is dealt with in an uncoupled way in this section in order to reduce the scales of algebraic systems to be solved. To be more specific, DCDE is solved by the Gaussian elimination method for given $x_{i,j}$ and $y_{i,j}$ $(i = 0, \dots, n;\ j = 0, \dots, m)$, and (26) and (29) are solved by the Newton–Raphson continuation method described in §3 for given $u_{i,j}$ $(i = 0, \dots, n;\ j = 0, \dots, m)$. It should be noticed that during the solution of (26) and (29), an interpolation procedure must be used for u because the computed values of u are given only on the previous mesh $(x_{i,j}, y_{i,j})$ $(i = 0, \dots, n;\ j = 0, \dots, m)$. Routines E02SAF and E02SBF from the NAG library are used for this purpose.



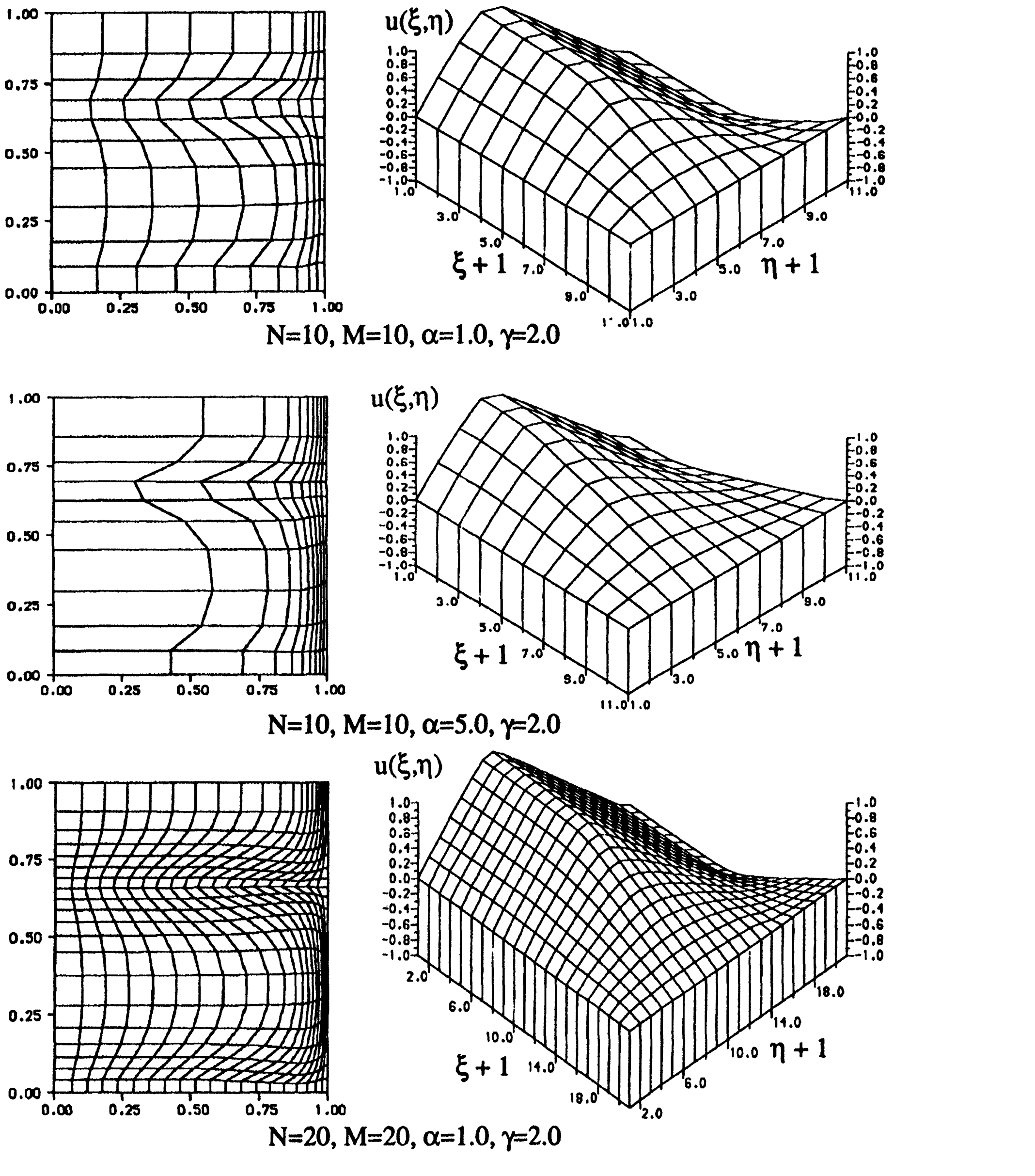
N=10, M=10, α=0.0, γ=0.0

N=10, M=10, α=1.0, γ=0.0

FIG. 3.3. *Solution and meshes for Example* 3.1 *with* $R = 15.0$, $\omega = 1.5\pi$.

In the solution procedure described below the subscripts i and j take the values $0 \le i \le n$ and $0 \le j \le m$. The process is initiated with the continuation parameter, α, set to zero and $u_{i,j}$ given by the solution of DCDE on a uniform mesh. α is increased by increments until it reaches the preselected value that is displayed with each set of results. At each value of α, the complete system is solved iteratively for $(x_{i,j}, y_{i,j})$ and $u_{i,j}$, and the superscript it denotes the iteration number. The algorithm may be written as:

(i) Set it:=0. Initial values $(x_{i,j}^{(0)}, y_{i,j}^{(0)})$ are values on a uniform mesh and initial values $u_{i,j}^{(0)}$ are given by solution of DCDE on the uniform mesh.



N=10, M=10, α=1.0, γ=2.0

N=10, M=10, α=5.0, γ=2.0

N=20, M=20, α=1.0, γ=2.0

FIG. 3.3. *Continued.*

(ii) Based on current values $u_{i,j}^{(it)}$, solve (26) and (29) by homotopy and a Newton–Raphson method (α serves as the continuation parameter). The previous mesh $(x_{i,j}^{(it)}, y_{i,j}^{(it)})$ is used to initiate the continuation process. The new mesh obtained in this step is denoted by $(x_{i,j}^{(it+1)}, y_{i,j}^{(it+1)})$.

(iii) Solve DCDE on the updated mesh $(x_{i,j}^{(it+1)}, y_{i,j}^{(it+1)})$ to obtain $u_{i,j}^{(it+1)}$.

(iv) Check convergence criterion on $u_{i,j}^{(it+1)}$. If this is not satisfied, then set $it := it + 1$ and return to (ii).

(v) Obtain final mesh by solving (26) and (29).

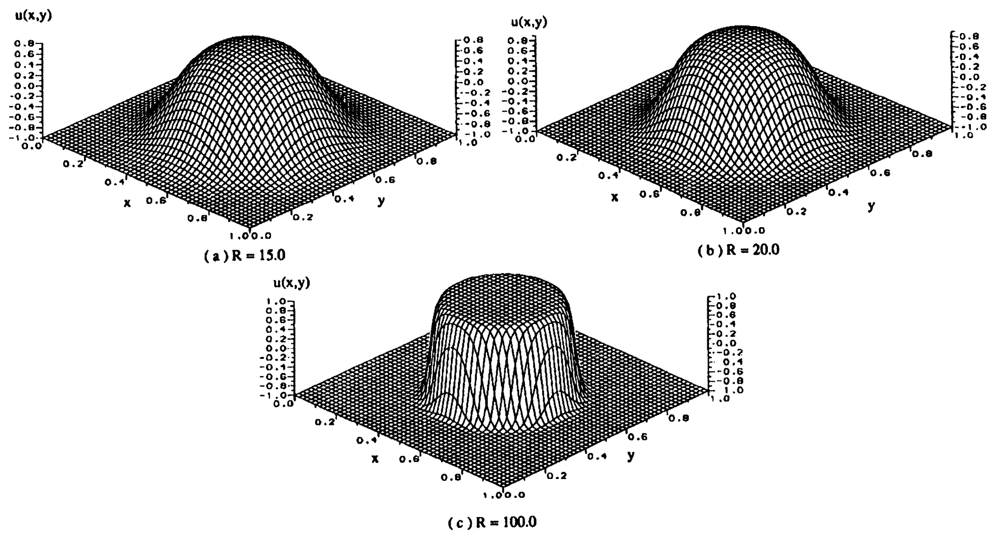


FIG. 3.4. *Exact solutions of Example* 3.2 *for different values of R.*

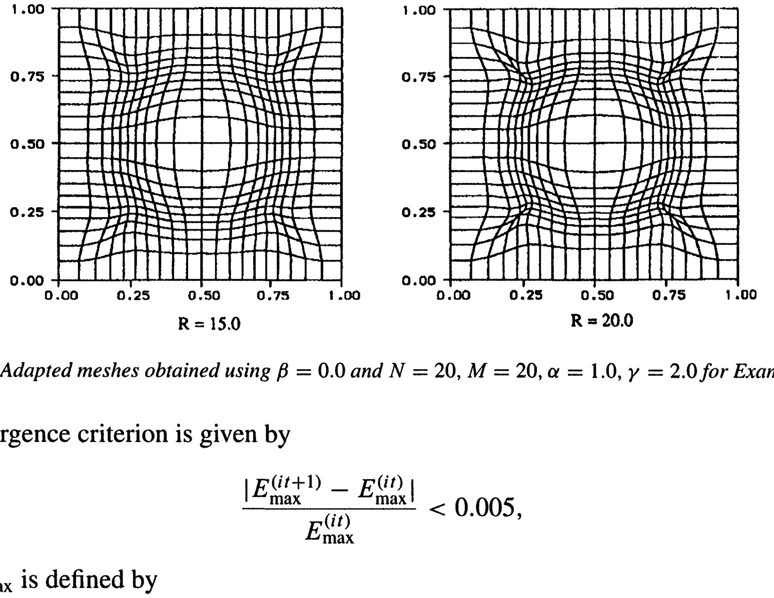


FIG. 3.5. *Adapted meshes obtained using* $\beta = 0.0$ *and* $N = 20$, $M = 20$, $\alpha = 1.0$, $\gamma = 2.0$ *for Example* 3.2.

The convergence criterion is given by

$$\frac{|E_{\max}^{(it+1)} - E_{\max}^{(it)}|}{E_{\max}^{(it)}} < 0.005, \tag{34}$$

where $E_{\max}$ is defined by

$$E_{\max} = \frac{1}{1 - e^{-R}} \max_{i,j} |u_{i,j} - u^e(x_{i,j}, y_{i,j})|. \tag{35}$$

Three cases have been considered in this section. In these cases, the parameter β is set to zero (without scaling).

Case 1. $R = 15$, $\omega = \pi$. The convergence histories, solutions at $y = 0.5$, and adapted meshes for several values of α and γ are shown in Fig. 4.1(a) and (b), respectively. It is observed easily from these figures that more accurate solutions can be obtained and the boundary layer can be resolved more sharply by grid adaption. These figures also show that the accuracy of the computed solution on a smooth mesh is better than that on an unsmooth mesh ($\alpha = 1.0$, $\gamma = 0.0$). By comparing Fig. 4.1(b) with Fig. 3.1, it can be seen that nearly the same mesh is obtained for given analytic function u and for computed solution $u_{i,j}$.

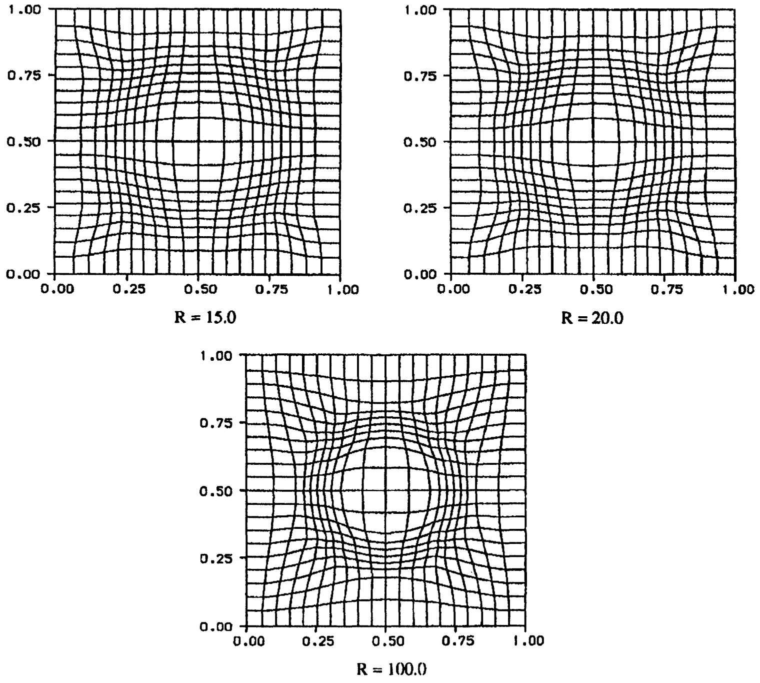


FIG. 3.6. *Adapted meshes obtained using* $\beta = 0.01$ *and* $N = 20$, $M = 20$, $\alpha = 1.0$, $\gamma = 2.0$ *for Example* 3.2.

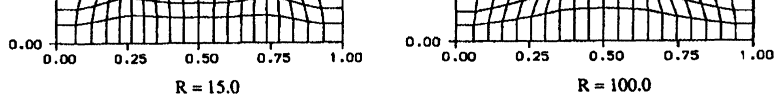


FIG. 3.7. *Adapted meshes obtained using* $\beta = 0.005$ *and* $N = 20$, $M = 20$, $\alpha = 1.0$, $\gamma = 2.0$ *for Example* 3.2.

Case 2. $R = 35$, $\omega = \pi$. In this case, the solution has a sharper boundary layer near $x = 1$. Figure 4.2 shows the convergence history, solutions at $y = 0.5$, and converged adapted mesh for $\alpha = 1.0$ and $\gamma = 2.0$. This figure consolidates the observations made for Case 1.

Case 3. $R = 15, \omega = 1.5\pi$. By comparing the graphs of exact solutions (Figs. 3.1–3.3), it is evident that this case is more difficult than the previous two cases because more adaption in the y direction is required. Figure 4.3 shows that although a less accurate solution is obtained for this case than for Cases 1 and 2, a more accurate solution is obtained on an adapted mesh than on a uniform mesh. It is also interesting to note in Fig. 4.3 that the solution obtained on the adapted mesh is less accurate than that obtained on a uniform mesh at points far from the boundary layer ($x \leq 0.7$) along $y = 0.5$, but the adapted mesh still gives a solution with much better resolution of the boundary layer.

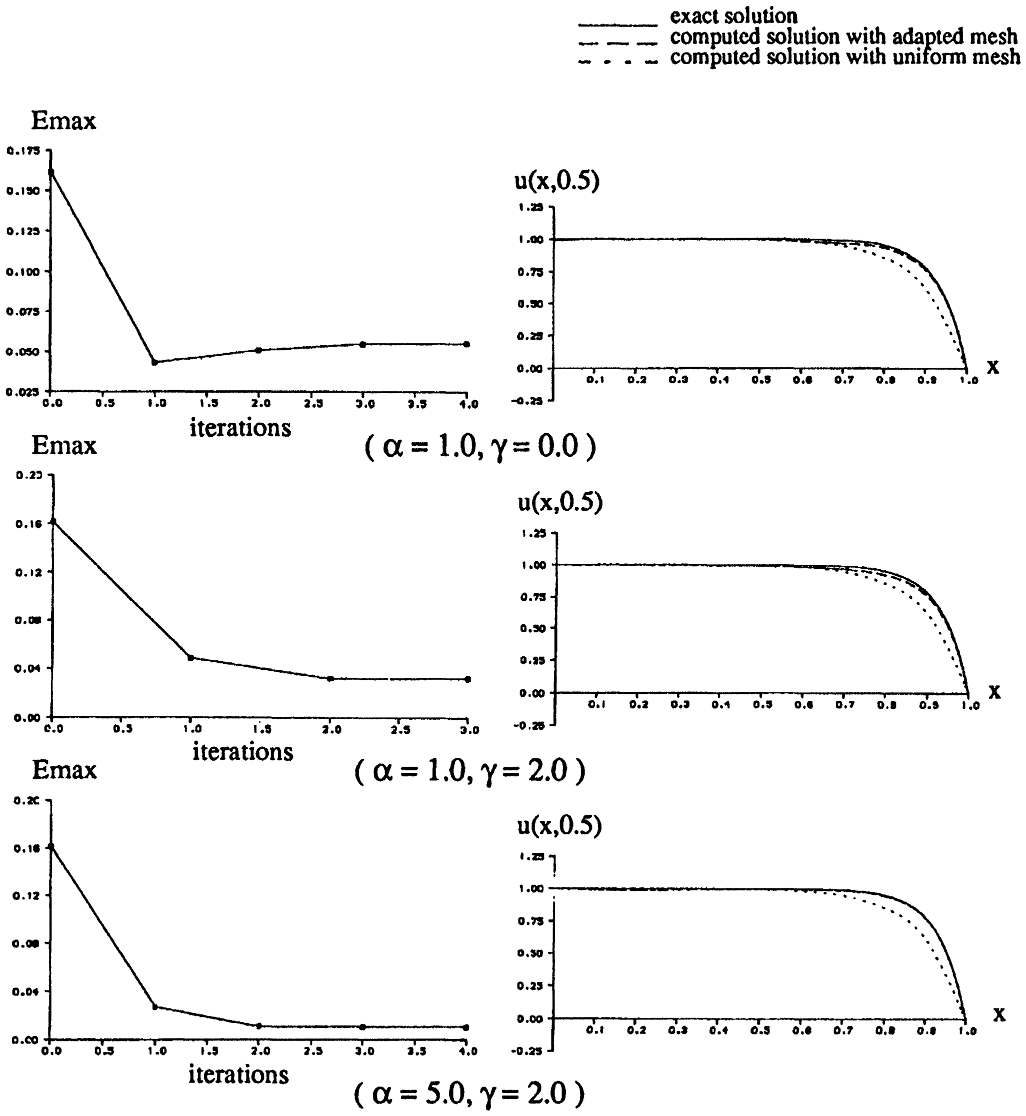
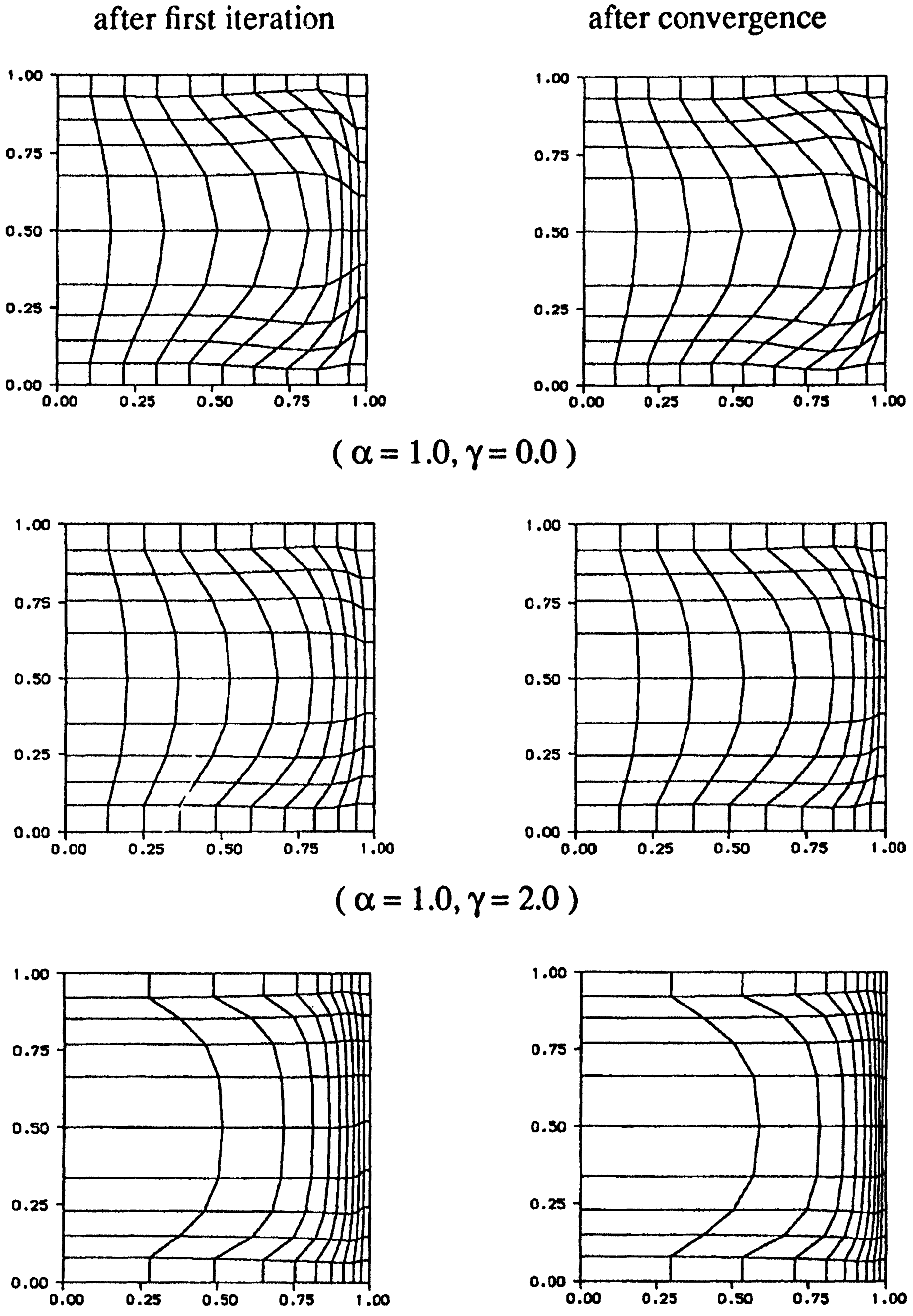


FIG. 4.1(a). *Convergence histories and solutions at* $y = 0.5$ *for Example* 4.1 ($R = 15.0, \omega = \pi$) *with* $N = 10, M = 10$.

Cases 1 and 3 have been considered by Dvinsky [6]. Indeed, our convergence criterion (34) has been chosen as in [6] to enable us to make comparisons with Dvinsky's work in terms of accuracy and efficiency. We notice that the present method gives results that are at least as accurate as those obtained by Dvinsky's method. But more flexibility is available in the present method, and much better results can be obtained by suitable choices of the parameters α, β, and γ. For example, as may be seen in Fig. 4.1, larger values of α can lead to better results.

5. Conclusions and comments. An understanding of the equidistribution principle is described and it is found that the equidistribution principle cannot be satisfied simultaneously on the whole structured mesh. Based on this understanding, a local (coordinate-line-based)


after first iteration

after convergence

(α = 1.0, γ = 0.0)

(α = 1.0, γ = 2.0)

(α = 5.0, γ = 2.0)

FIG. 4.1(b). *Adapted meshes for Example* 4.1 ($R = 15.0$, $\omega = \pi$) *with* $N = 10$, $M = 10$.

equidistribution principle is developed and a simple adaptive grid method with three parameters α (concentration), β (scaling), and γ (smoothness) is presented.

Numerical experiments have been performed on grid adaption, both for given analytic function and for numerical solutions of partial differential equations in two dimensions. They

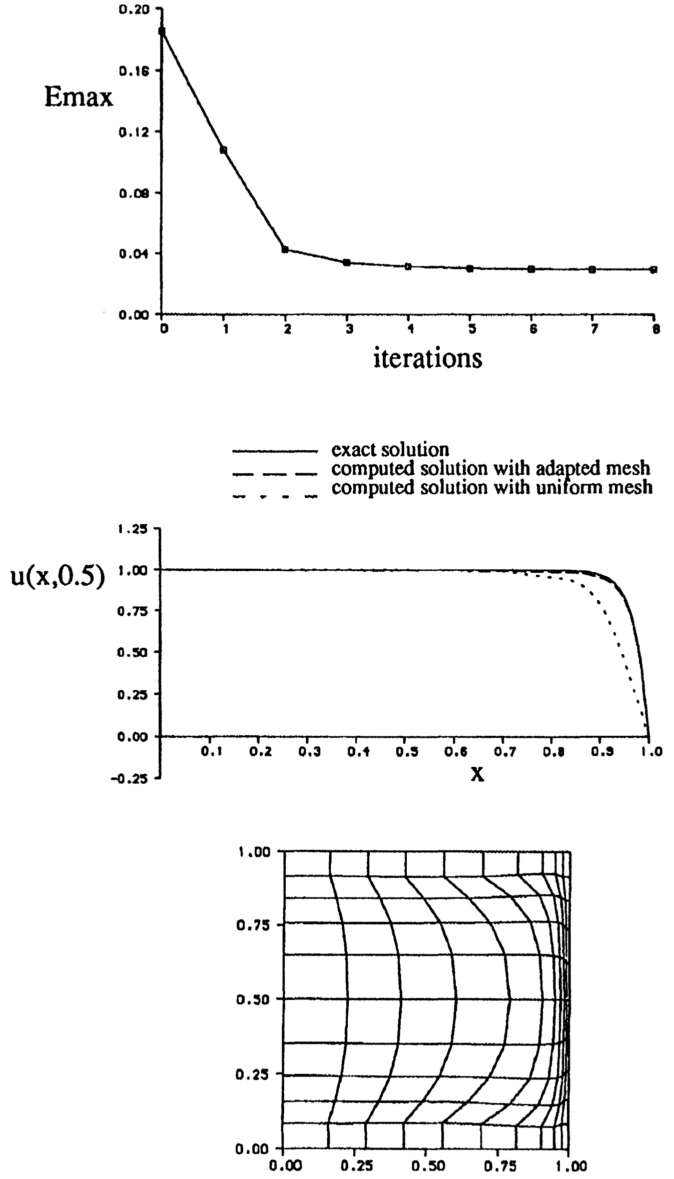


FIG. 4.2. *Convergence history, solutions at* $y = 0.5$ *and adapted mesh for Example* 4.1 ($R = 35.0, \omega = \pi$) *with* $N = 10, M = 10, \alpha = 1.0, \gamma = 2.0$.

show that the present adaptive grid method is fairly insensitive to the choice of values of parameters and it can lead to much better results than those obtained on uniform meshes. The equidistribution principle presented here has only been applied to a simple, steady, two-dimensional convection-diffusion equation. It should be possible, however, to apply the multidimensional adaptive grid method to time-dependent problems. A step in this direction has recently been made by Huang, Ren, and Russell [11], [12] with their moving mesh partial differential equations based on versions of the one-dimensional equidistribution condition (16). The moving mesh equations control the locations of the nodes in such a way that the equidistribution condition remains satisfied as time evolves.

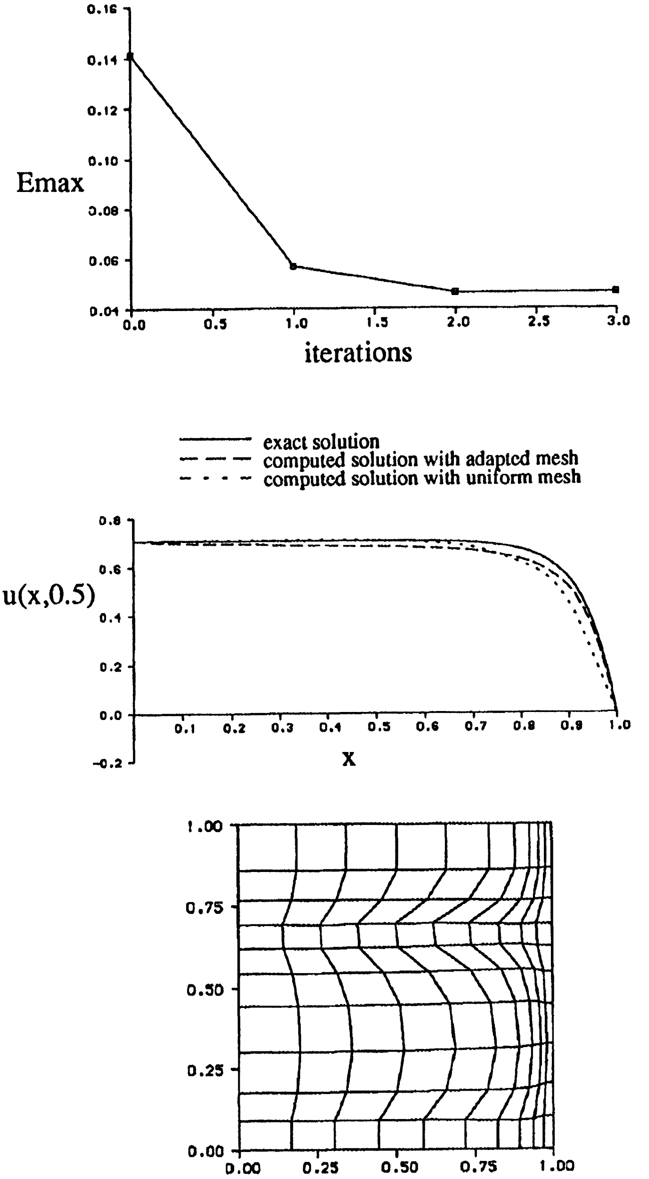


FIG. 4.3. *Convergence history, solutions at* $y = 0.5$ *and adapted mesh for Example* 4.1 $(R = 15.0, \omega = 1.5\pi)$ *with* $N = 10, M = 10, \alpha = 1.0, \gamma = 2.0$.

Obviously, this method needs further investigation and it also needs to be applied to more problems. The efficiency of the proposed method relative to fixed-grid methods is also an area which has to be investigated. At this stage we can only say that for problems with steep gradients, adaptive methods are usually better in terms of computational efficiency. The reader is referred to the work by Hawken, Gottlieb, and Hansen [10] for further commentary on this matter. We hope that our presentation of the equidistribution principle in multidimensions may be useful for the development of other new adaptive grid methods.

Acknowledgements. The authors wish to thank the referees for their helpful comments.

REFERENCES

[1] J. U. Brackbill and J. S. Saltzman, *Adaptive zoning for singular problems in two dimensions*, J. Comput. Phys., 46 (1982), pp. 342–368.

[2] M. O. Bristeau, R. Glowinski, L. Dutto, J. Périaux, and G. Rogé, *Compressible viscous flow calculations using compatible finite element approximations*, Internat. J. Numer. Methods Fluids, 11 (1990), pp. 719–749.

[3] D. Catherall, *The adaption of structured grids to numerical solutions for transonic flow*, Internat. J. Numer. Methods Engrg., 32 (1991), pp. 921–937.

[4] C. de Boor, *Good approximation by splines with variable knots* II, in Lecture Notes in Mathematics 363, Springer-Verlag, New York, 1974, pp. 12–20.

[5] E. A. Dorfi and L. O'C. Drury, *Simple adaptive grids for* 1-D *initial value problems*, J. Comput. Phys., 69 (1987), pp. 175–195.

[6] A. S. Dvinsky, *Adaptive grid generation from harmonic maps on Riemannian Manifolds*, J. Comput. Phys., 95 (1991), pp. 450–476.

[7] H. A. Dwyer, R. J. Kee, and B. R. Sanders, *Adaptive grid method for problems in fluid mechanics and heat transfer*, AIAA J., 18 (1980), pp. 1205–1212.

[8] H. A. Dwyer, *Grid adaption for problems in fluid dynamics*, AIAA J., 22 (1984), pp. 1705–1712.

[9] R. M. Furzland, J. G. Verwer, and P. A. Zegeling, *A numerical study of three moving-grid methods for* 1-D PDEs *which are based on the method of lines*, J. Comput. Phys., 89 (1990), pp. 349–388.

[10] D. F. Hawken, J. J. Gottlieb, and J. S. Hansen, *Review of some adaptive node-movement techniques in finite element and finite difference solutions of* PDEs, J. Comput. Phys., 95 (1991), pp. 254–302.

[11] W. Huang, Y. Ren, and R. D. Russell, *Moving mesh* PDEs *based on the equidistribution principle*, SIAM J. Numer. Anal., 31 (1994), pp. 709–730.

[12] ———, *Moving mesh methods based on moving mesh* PDEs, J. Comput. Phys., submitted.

[13] J. Peraire, M. Vahdati, K. Morgan, and O. C. Zienkiewicz, *Adaptive remeshing for compressible computations*, J. Comput. Phys., 72 (1987), pp. 449–466.

[14] J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin, *Numerical Grid Generation*, North-Holland, New York, 1985.

[15] A. Winslow, *Numerical solution of the quasilinear Poisson equation in a nonuniform triangular mesh*, J. Comput. Phys., 2 (1967), pp. 149–172.