INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 13, 1085–1111 (1991)

A LOCAL ADAPTIVE GRID PROCEDURE FOR INCOMPRESSIBLE FLOWS WITH MULTIGRIDDING AND EQUIDISTRIBUTION CONCEPTS

F. MOUKALLED AND S. ACHARYA*

Mechanical Engineering Department, Louisiana State University, Baton Rouge, LA 70803, U.S.A.

SUMMARY

An adaptive grid solution procedure is developed for incompressible flow problems in which grid refinement based on an equidistribution law is performed in high-error-estimate regions that are flagged from a preliminary coarse grid solution. Solutions on the locally refined and equidistributed meshes are obtained using boundary conditions interpolated from the preliminary coarse grid solution, and solutions on both the refined and coarse grid regions are successively improved using a multigrid approach. For this purpose, suitable correction terms for the coarse grid equations are derived for all variables in the flagged regions. This procedure with Local Adaptation, Multigridding and Equidistribution (LAME) concepts is applied to various flow problems to demonstrate the accuracy improvements obtained using this method.

KEY WORDS Adaptive grid Multigridding Equidistribution Fluid flow Improved accuracy

INTRODUCTION

This paper deals with the development of a computationally efficient solution-adaptive grid procedure for incompressible flows. In a solution-adaptive grid procedure the grid points move dynamically in response to the evolving solution in order to equidistribute a measure of the numerical error. Thus, as desired, grid points are always densely clustered in regions of high error estimates.

The literature dealing with solution-adaptive grid procedures has recently been reviewed by Thompson.[1] In general, these methods can be broadly classified into two groups which will be referred to as the global refinement method and the local refinement method in this paper. Both methods attempt to refine the grid in the important regions but do so in different ways.

In the global refinement method[2–11] all grid points participate in the adaptation process. A weighting function W_{ij} which is proportional to a measure of the error is calculated at all points and the mesh size Δs_{ij} adjusted such that $W_{ij} \cdot \Delta s_{ij}$ is nearly equal at every point in the domain. Thus, as desired, the mesh spacing Δs_{ij} is small where the error measure W_{ij} is large and therefore the resulting solution error is nearly equidistributed. In References 2–5 either the first or second derivative of the dependent variable or a combination of the two is used as the weighting function (or measure of the error). In Reference 5 the analogy between the driving forces represented by the weighting function and the tensional spring force was developed; this was extended by Nakahashi

* Author to whom all correspondence should be addressed.

0271–2091/91/191085–27$13.50

Received 11 February 1990
Revised 16 November 1990

and Deiwert,[6] who introduced the idea of a torsional spring attached to each node in order to control grid skewness. Rai and Anderson,[7,8] Greenberg[9] and Eiseman[10] have proposed grid point movement to be controlled by forcing or weighting functions that either attract or repel grid points relative to each other. Recently Kim and Thompson[11] have considered two different approaches to global grid point redistribution. In the first approach the control functions in the elliptic Poisson equations for the grid are used to drive grid point movement by relating the control function to the pressure gradient. In the second approach a variational technique is used.

In the local refinement method[12-17] a somewhat different approach towards the same end is adopted which involves refinement only in certain subregions. Berger and Jameson[12] and Phillips and Schmidt[13] both advocate flagging rectangular regions with high truncation error estimates calculated from a preliminary solution and then uniformly subdividing the flagged regions. A more accurate solution is then obtained in the flagged regions. Berger and Oliger[14] and Skamarock *et al.*[15] follow a similar approach to that in Reference 12 but allow for multiple, rotated, overlapping fine grids and update the coarse grid solution by the appropriately interpolated fine grid values at each step. Thompson and Ferziger[16] and Acharya and Moukalled[17] have both incorporated the ideas of multigrid calculations together with adaptive gridding. In Reference 16 rectangular regions are flagged from a preliminary error estimate; the meshes in the flagged regions are then uniformly refined and multigrid calculations performed between the various grid levels. In Reference 17 a versatile procedure is presented with Local Adaptation, Multigridding and Equidistribution (LAME) concepts. Unlike other related studies, the LAME procedure permits the flagging of an arbitrarily shaped region in the computational domain; further, instead of refining the flagged region uniformly as in most other studies, it performs the refinement using an error equidistribution method. Thus each flagged region is refined in an optimal fashion. Various levels of flagged regions can be overlayed on the preliminary grid and multigrid-type calculations are performed until the desired level of accuracy is obtained.

The LAME procedure has been presented in Reference 17 only for convection–diffusion-type problems. In this paper the method is extended to flow problems. This extension involves a number of important issues which are discussed in this paper. For example, to avoid checkerboard pressure and velocity fields, a staggered grid is normally used for the velocity components. Adaptation on such a grid, therefore, implies redefining three different sets of grid points at each step. This can be very expensive and is an issue specific to flow field calculations. Other examples of issues that are specific to flow calculations include the derivation of corrector (or restriction) terms for the pressure Poisson equation (which is based on the continuity equation) and the need for conservative mass flux interpolation across the flagged region boundaries. These issues are discussed in detail in this paper.

In the following sections the calculation procedure on a non-staggered grid is briefly described[18] followed by a brief review of the LAME solution-adaptive grid methodology for convection–diffusion problems.[17] The extension of the LAME procedure to flow problems is described next followed by the application of this procedure on a number of test problems.

CALCULATION ON A NON-STAGGERED CURVILINEAR GRID

The solution of flow equations (conservation of mass and momentum equations) using SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) algorithms[19] is generally performed on a staggered grid to avoid checkerboard pressure and velocity fields. Thus in a two-dimensional flow field three sets of mesh positions (one each for the two components of velocity and the third for pressure) and associated metric quantities have to be calculated and stored. If grid adaptation is

performed, the grid changes at every step and therefore three sets of grid positions and metric quantities have to be calculated at each step. This makes the process computationally very intensive. To avoid this, a non-staggered grid arrangement is desirable, but the solution algorithm must be suitably modified to avoid checkerboard pressure and velocity fields. Such modifications have been presented in a few studies including those of Hsu,[20] Reggio and Camarero,[21] Rhie and Chow[22] and Acharya and Moukalled.[18]

In this paper the modified algorithm called SIMPLEM (SIMPLE Modified) proposed by Acharya and Moukalled[18] will be used. In Reference 18 the SIMPLEM algorithm has been compared with some of the other algorithms (e.g. Reference 22) and shown to have superior characteristics. For these reasons the SIMPLEM algorithm is used in this paper and is described briefly later in this section.

Attention is now turned to a very brief description of the various steps in the calculation process. These steps are: grid generation, discretization of the conservation equations and the SIMPLEM solution algorithm. Only a brief discussion is given here since many of these details are documented elsewhere (see e.g. References 17–19 and 23).

Grid generation

The computational grid (ξ, η) is generated in an arbitrary physical domain by solving a system of Poisson equations[23]

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = -J^2(x_\xi P^* + x_\eta Q^*), \tag{1}$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = -J^2(y_\xi P^* + y_\eta Q^*), \tag{2}$$

where α, β, γ and J are metric quantities given by

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

and P^* and Q^* are control functions that can be chosen to provide denser clustering in certain specified areas.

Anderson and Steinbrenner[24] have recast equations (1) and (2) into an error equidistribution law by relating P^* and Q^* to the local error estimate or weighting function. To do this, equations (1) and (2) are re-expressed by defining $\tilde{\tau} = J^2 P^*/\alpha$ and $\tilde{\psi} = J^2 Q^*/\gamma$, then substituting into equations (1) and (2) and finally eliminating $\tilde{\psi}$ between the two equations. This leads to an equation that can be expressed with s_1, the arc length along a line of constant η, as the dependent variable:

$$(s_1)_{\xi\xi} + \tau_1 (s_1)_\xi = 0, \tag{4}$$

where

$$\tau_1 = \tilde{\tau} - [(\theta_1)_\xi - 2(\theta_2)_\xi]\cot\theta_3 - (s_1)_\xi (\theta_2)_\eta \sin\theta_3/(s_2)_\xi. \tag{5}$$

In the above equation θ_1 and θ_2 are the slopes of constant η and ξ curves and θ_3 is the angle of intersection. A similar equation can be derived for s_2, the arc length along a line of constant ξ:

$$(s_2)_{\eta\eta} + \psi_1 (s_2)_\eta = 0. \tag{6}$$

If these equations for s_1 and s_2 are compared with the equidistribution laws along constant η and ξ lines, the relationship between τ_1 and ψ_1 (and therefore P^* and Q^*) and the weighting functions W can be established. For example, along a constant η line the equidistribution law is

$$(s_1)_\xi W = f_1(\eta),$$

which when differentiated with respect to ξ leads to

$$(s_1)_{\xi\xi}+(W_\xi/W)(s_1)_\xi=0. \tag{7}$$

Comparing equation (7) with equation (4) leads to

$$\tau_1=W_\xi/W; \tag{8}$$

similarly,

$$\psi_1=W_\eta/W. \tag{9}$$

Thus the solution of the conventional elliptic grid generation equations (1) and (2) with P^* and Q^* obtained from equations (8) and (9) leads to an equidistributed grid. This is the procedure adopted in this paper for performing grid adaptation during the calculation. The weighting functions W are a measure of the local error; how they are calculated is described later in this paper.

Discretization of the conservation equations

The conservation of x- or y-momentum can be expressed as

$$\{\rho G_1\phi-(\Gamma/J)(\alpha\phi_\xi-\beta\phi_\eta)\}_\xi+\{\rho G_2\phi-(\Gamma/J)(\gamma\phi_\eta-\beta\phi_\xi)\}_\eta=-J\nabla p+JS, \tag{10}$$

where the dependent variable ϕ denotes either u or v, G_1/J and G_2/J are the contravariant velocity components with G_1 and G_2 defined by

$$G_1=uy_\eta-vx_\eta, \qquad G_2=vx_\xi-uy_\xi, \tag{11}$$

and S, ρ and Γ denote respectively the source term, the fluid density and the diffusion coefficient.

In order to reduce equation (10) to a discretization equation, the control volume approach is adopted in which the domain is subdivided into control volumes, each containing a grid point, as shown in Figure 1. Equation (10) is integrated on a typical control volume around grid point P (see Figure 1). This leads to an integral balance equation of the form

$$\begin{aligned}&[\rho G_1\phi-(\Gamma/J)(\alpha\phi_\xi-\beta\phi_\eta)]_e\Delta\eta-[\rho G_1\phi-(\Gamma/J)(\alpha\phi_\xi-\beta\phi_\eta)]_w\Delta\eta\\&+[\rho G_2\phi-(\Gamma/J)(\gamma\phi_\eta-\beta\phi_\xi)]_n\Delta\xi-[\rho G_2\phi-(\Gamma/J)(\gamma\phi_\eta-\beta\phi_\xi)]_s\Delta\xi\\&=S_P J\Delta\eta\Delta\xi-(y_\eta p_\xi-y_\xi p_\eta)_P\Delta\xi\Delta\eta.\end{aligned} \tag{12}$$

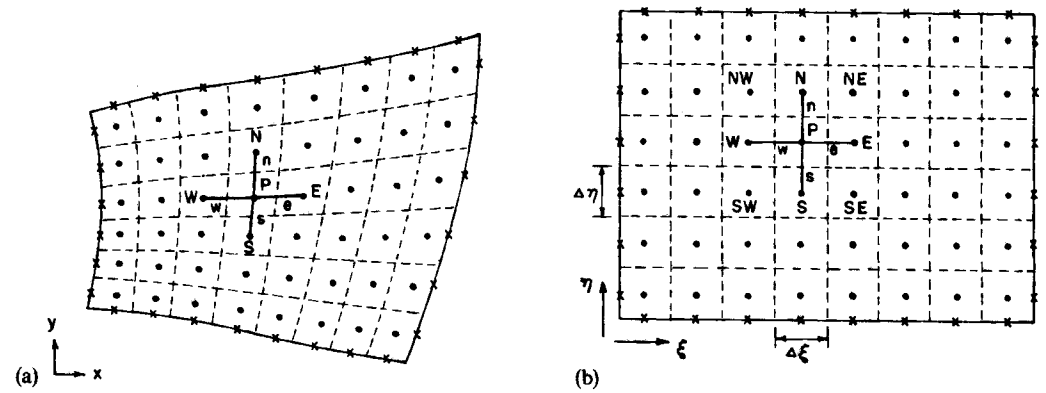

Figure 1. Curvilinear co-ordinate system in (a) physical space and (b) computational space

In the above equation ϕ, ϕ_ξ and ϕ_η at the control volume interfaces (e, w, n and s) are expressed in terms of the grid point values by using the power law approximation,[19] which has been shown to be superior to the central difference, the upwind and hybrid schemes. The resulting algebraic equation for u has the form

$$a_P^u u_P = a_E^u u_E + a_W^u u_W + a_N^u u_N + a_S^u u_S + b_0^u + b_{no}^u + (b_1^u p_\xi + c_1^u p_\eta)_P, \tag{13}$$

where a_P^u, a_E^u, a_W^u, a_N^u and a_S^u are the convection–diffusion coefficients for u, b_0^u is the source term contribution, b_{no}^u is the contribution due to non-orthogonality and the last two terms denote the pressure gradient contribution. A similar equation can be derived for the v-velocity.

Equation (13) can be re-expressed by dividing across by a_P^u and denoting the resulting coefficients on the right-hand side (RHS) of the equation by uppercase letters. The equation can then be written as

$$u_P = \sum_{n=E,W,N,S} A_n^u u_n + B_0^u + B_{no}^u + (B_1^u p_\xi + C_1^u p_\eta)_P \tag{14}$$

or

$$u_P = \hat{u}_P + (B_1^u p_\xi + C_1^u p_\eta)_P, \tag{15}$$

where $\hat{u}_P$ represents the first three terms of the RHS of equation (14). Similarly,

$$v_P = \hat{v}_P + (B_1^v p_\xi + C_1^v p_\eta)_P. \tag{16}$$

In the SIMPLEM algorithm of Acharya and Moukalled,[18] p_ξ and p_η at grid point P in equations (15) and (16) are derived using a $2-\delta\xi$ and a $2-\delta\eta$ centred difference scheme. Introducing equations (15) and (16), i.e. $u=\hat{u}+(B_1^u p_\xi + C_1^u p_\eta)$ and $v=\hat{v}+(B_1^v p_\xi + C_1^v p_\eta)$, in the definitions of the contravariant terms G_1 and G_2, the following equations can be obtained:

$$G_1 = \hat{G}_1 + (B_1 y_\eta - B_2 x_\eta) p_\xi + (C_1 y_\eta - C_2 x_\eta) p_\eta, \tag{17}$$

$$G_2 = \hat{G}_2 + (C_2 x_\xi - C_1 y_\xi) p_\eta + (B_2 x_\xi - B_1 y_\xi) p_\xi, \tag{18}$$

where

$$\hat{G}_1 = \hat{u} y_\eta - \hat{v} x_\eta, \qquad \hat{G}_2 = \hat{v} x_\xi - \hat{u} y_\xi. \tag{19}$$

Denoting

$$\hat{B}_1 = B_1 y_\eta - B_2 x_\eta, \qquad \hat{B}_2 = B_2 x_\xi - B_1 y_\xi$$

and, similarly,

$$\hat{C}_1 = C_1 y_\eta - C_2 x_\eta, \qquad \hat{C}_2 = C_2 x_\xi - C_1 y_\xi,$$

equations (17) and (18) can be written in an abbreviated form as

$$G_1 = \hat{G}_1 + \hat{B}_1 p_\xi + \hat{C}_1 p_\eta, \qquad G_2 = \hat{G}_2 + \hat{C}_2 p_\eta + \hat{B}_2 p_\xi. \tag{20}$$

To derive the pressure equation, G_1 and G_2 from equation (20) are introduced into the discretized continuity equation given by

$$(\rho G_1 \Delta\eta)_e - (\rho G_1 \Delta\eta)_w + (\rho G_2 \Delta\xi)_n - (\rho G_2 \Delta\xi)_s = 0. \tag{21}$$

Using centred differences for p_ξ and p_η, the following equation for pressure is obtained from equation (21):

$$a_P p_P = \sum_{n=E,W,N,S} a_n p_n + b_P + b_{no}, \tag{22}$$

where

$$a_E = -(\rho \hat{B}_1 \Delta\eta/\delta\xi)_e, \quad a_W = -(\rho \hat{B}_1 \Delta\eta/\delta\xi)_w, \quad a_N = -(\rho \hat{C}_2 \Delta\xi/\delta\eta)_n, \quad a_S = -(\rho \hat{C}_2 \Delta\xi/\delta\eta)_s, \tag{23}$$

$$a_P = a_E + a_W + a_N + a_S, \tag{24}$$

$$b_P = (\rho \hat{G}_1 \Delta\eta)_w - (\rho \hat{G}_1 \Delta\eta)_e + (\rho \hat{G}_2 \Delta\xi)_s - (\rho \hat{G}_2 \Delta\xi)_n, \tag{25}$$

$$b_{no} = (\hat{C}_1 p_\eta)_w - (\hat{C}_1 p_\eta)_e + (\hat{B}_2 p_\xi)_s - (\hat{B}_2 p_\xi)_n. \tag{26}$$

Solution algorithm

The SIMPLEM algorithm consists of the following steps.

1. Start with guessed fields u^* and v^*.
2. Calculate the coefficients of the momentum equations and then determine $\hat{u}$ and $\hat{v}$. From these values determine $\hat{G}_1$ and $\hat{G}_2$ at the nodes and interpolate linearly to find $\hat{G}_1$ and $\hat{G}_2$ at the interfaces.
3. Calculate the coefficients of the pressure equation (equation (22)) and solve the resulting system of equations to get a new pressure field.
4. Update G_1 and G_2 at the interfaces (equations (17) and (18)) using centred differences with the new pressure field.
5. Recalculate the momentum equation coefficients and solve the momentum equations to get a new velocity field.
6. Return to step 2 and repeat until a converged solution is obtained.

It was shown in Reference 18 that SIMPLEM was robust and efficient and effectively suppressed checkerboard pressure and velocity fields.

THE LAME SOLUTION ADAPTIVE GRID PROCEDURE FOR FLOW PROBLEMS

In this section the adaptive grid procedure developed here for flow problems is described. The procedure is an extension of the LAME procedure[17] developed earlier by the authors for convection–diffusion equations. However, solution of the flow equations in an adaptive grid framework involves additional issues.

In the adaptive grid procedure developed in this paper the grid is refined locally in flagged regions where the error estimate exceeds a specified threshold value. Compared to other reported studies, the present method has the advantage that the flagged region in the computational space where the preliminary calculations are being performed can be arbitrarily shaped (and not necessarily rectangular or made up of overlapping rectangles). More importantly, grid refinement in each flagged region is not done uniformly as in other studies but by an equidistribution law that clusters the points more densely in regions of higher error estimates within the flagged region. The solutions at different grid levels are successively improved using a multigrid method. In the discussion that follows, the local grid refinement using error equidistribution laws is described first followed by a description of the multigrid flow calculation procedure. Issues specific to flow calculations are included in this discussion.

Local grid refinement using error equidistribution law

Grid refinement in this procedure is based on error estimates calculated from a preliminary coarse grid solution. These error estimates are used firstly to flag important regions where the

mesh should be refined and secondly to guide the distribution of grid points in these regions using an error equidistribution law.

The first step is calculating the error estimate for a dependent variable ϕ. This is done from a preliminary coarse grid solution by defining a weighting function W_ϕ for each dependent variable as

$$W_\phi = \alpha_1 J|\nabla^2\phi| + \alpha_2|\nabla\phi|^2 + \alpha_3 J|\nabla\xi\cdot\nabla\eta| + \alpha_4 J[(\nabla\xi)^2 + (\nabla\eta)^2], \tag{27}$$

where α_1, α_2, α_3 and α_4 are constants. The first two terms represent a measure of the truncation error, since the error estimate at any point is proportional to $(\Delta s)^n f(\nabla\phi, \nabla^2\phi, \ldots, \nabla^n\phi)$, where n denotes the order of the differencing scheme. The last two terms in equation (27) represent grid orthogonality and smoothness and can be viewed as a measure of the geometric error arising from grid distortion. To define local regions where the grid should be refined, a normalized weighting function is defined as

$$\tilde{W}_\phi = (W_\phi + 1)/(W_{\phi,\max} + 1) \tag{28}$$

and points are flagged if the normalized weighting function is greater than a preassigned value (typically 0·3–0·5). A flagged region is identified as a cluster of contiguous flagged points. There can be more than one such cluster and each cluster in the ξ–η space can have an arbitrary shape. This is an important feature of this method and, unlike other competing methods, does not restrict the refinement to be only in rectangular regions.

In each flagged region the grid is refined using an error equidistribution method. This is done by doubling the number of grid points in the flagged region and solving the Poisson equations for grid generation, equations (1) and (2), but with the control functions P^* $(=\alpha\tilde{\tau}/J^2)$ and Q^* $(=\gamma\tilde{\psi}/J^2)$ related to the weighting functions or error measures W_ϕ through equations (8) and (9). The resulting grid is non-uniform with a finer mesh in the regions where the weighting function values are higher. This approach of *non-uniformly* refining the mesh using an equidistribution scheme, but only in locally important regions that can have an arbitrary shape in the computational domain, makes the present method very versatile and computationally efficient.

If different variables need clustering in different regions, the use of an error measure based on a single variable will not provide a grid that is optimal for every variable. To resolve this problem, at each point the error estimate for each variable is computed and the largest of these estimates is selected at that point. Having calculated the largest error estimate at each point, they are then normalized. This procedure permits the grid to be defined based on the needs of all dependent variables.

In using the equidistribution approach described above, the flagged region is essentially being mapped to a new computational space, say ξ'–η' (see Figure 2). In performing this mapping, the boundaries of the flagged region (shown shaded in Figure 2) are appropriately divided into four parts and assigned as the $\xi'_{\min}$, $\xi'_{\max}$, $\eta'_{\min}$ and $\eta'_{\max}$ boundaries. This is illustrated in Figure 2 with 1–2 and 4–3 corresponding to the $\xi'_{\min}$ and $\xi'_{\max}$ boundaries and 1–4 and 2–3 corresponding to the $\eta'_{\min}$ and $\eta'_{\max}$ boundaries. Once the new boundaries are assigned, the number of grid points along each boundary is doubled and a preliminary grid is generated in the flagged region by solving equations (1) and (2). The values of P^* and Q^* on this preliminary grid are interpolated from the corresponding coarse grid values of P^* and Q^* and equations (1) and (2) are then solved to obtain the equidistributed mesh in the flagged region. This procedure can be extended further and taken to convergence by again interpolating the values of P^* and Q^* onto the new equidistributed mesh and again solving equations (1) and (2). This process can be repeated until the grid points undergo no further change in position.

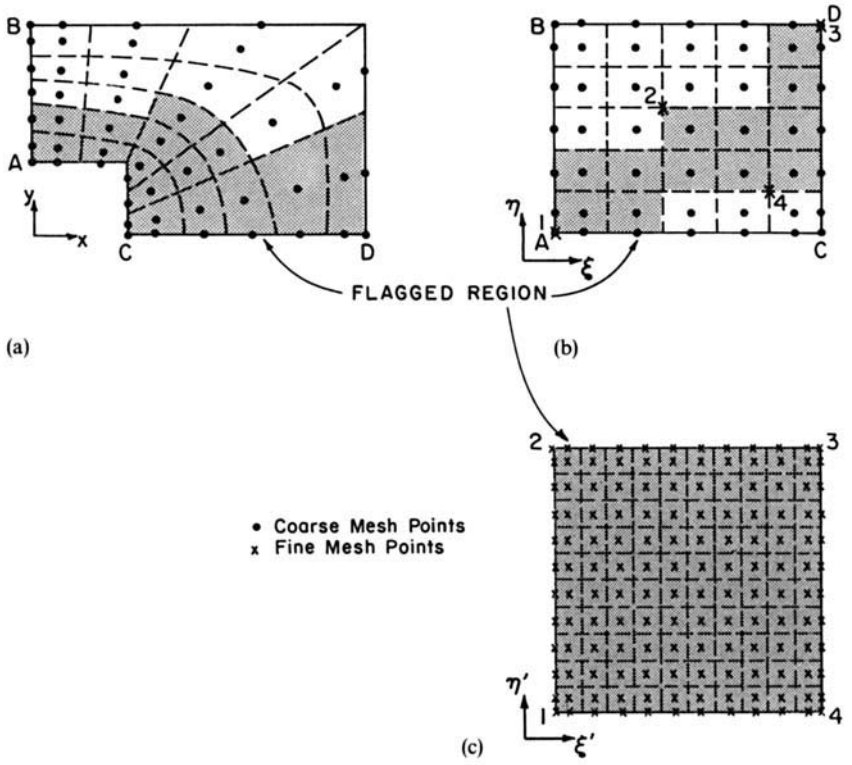

Figure 2. Typical flagged region (shown shaded) in (a) physical space, (b) ξ–η computational space and (c) ξ'–η' flagged computational space

The above procedure for generating an equidistributed grid is repeated in each flagged region. Once the equidistributed mesh in each flagged region is obtained, the next step of obtaining the solution in each of these regions is initiated.

Boundary conditions for the locally refined regions

The boundary conditions for the locally refined regions have to be interpolated from the coarser outer mesh in which the refined region is embedded. The accuracy of the solution in the refined region will depend on the accuracy of the interpolated boundary condition. Therefore both the solution accuracy on the outer mesh and the accuracy of the interpolation are important.

For flow problems it is important that the interpolation should be such that it conserves mass across the zonal boundaries. The interpolated mass flow rates are needed in the pressure equation for the boundary control volumes in each flagged region. A simple conservative interpolation procedure is developed in this work. Figure 3 shows a flagged region boundary (AB) that is transformed, for example, to a $\xi'_{\min}$ boundary. The boundary AB in the outer mesh is along both ξ- and η-co-ordinate lines. In the ξ'–η' space AB is along a line of constant ξ' and contains twice as many points, distributed non-uniformly, compared to the number of points along AB in the ξ–η space. To show how this interpolation is done, consider the portion AC of the boundary (Figure 3) with a flow rate F_n across it in the ξ–η space. In the ξ'–η' space the flow rate across a fine grid interface along AC is given by F'_k, where

$$F'_k = N_k F_n, \tag{29}$$

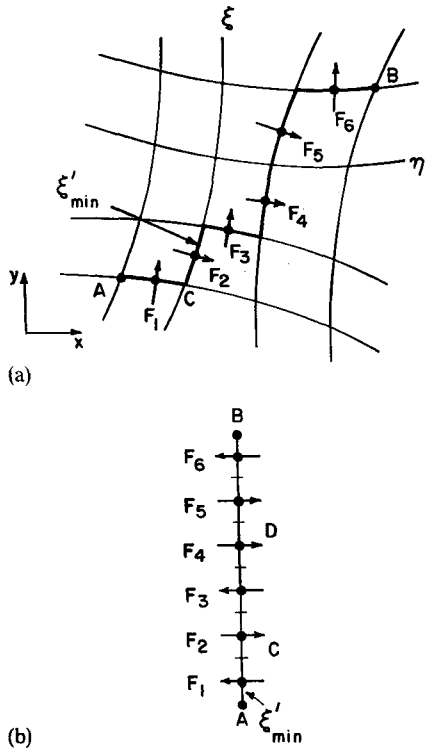

Figure 3. Boundary of flagged region in (a) physical space and (b) ξ'–η' computational space

and such that

$$\sum_k F'_k = \sum_k N_k F_n = F_n, \tag{30}$$

where N_k is the interpolating factor and for a simple linear interpolation is given by length ratios. Thus from equation (30) mass conservation is ensured during interpolation. In general, and as shown in Figure 3(b), if the boundary flow rates in the ξ–η space are denoted by $F_1, F_2, F_3, \ldots$, the interpolated flow rates in the ξ'–η' space can be written as

$$F'_k = \sum_j N_{kj} F_j, \tag{31}$$

where N_{kj} depends on the order of interpolation. Once the mass flow rates across the fine grid boundaries have been interpolated, the normal velocity (u, or v) is calculated by dividing the interpolated mass flow rates by the corresponding flow area and the density.

Within the flagged region, since a conservative differencing procedure is employed in discretizing the continuity equation (equation (21)), the converged velocity field and the velocity field updated after the solution of the pressure equation at each iteration always satisfy local continuity.

The tangential velocity component and other variables are interpolated by a simpler practice. The values at the corners of the coarse grid control volume faces along the flagged region boundaries are first calculated as the weighted average of the four neighbouring coarse grid points. Linear interpolation is then used between the coarse grid corner values to determine the fine grid point boundary values.

Multigrid calculations

As noted earlier, the solution accuracy in the flagged regions will also depend on the accuracy of the outer grid solution. To improve the solution accuracy on the outer mesh, a multigrid-type approach similar in its essential concept to the FAS (Full Approximation Storage) algorithm described by Brandt[25] and Vanka[26] is used in this paper. In the multigrid method, calculations are done on different grid levels with prolongation from the coarse grid to the fine grid and restriction from the fine grid to the coarse grid. As in Reference 26, the prolongation operation is performed by some form of interpolation, but in this paper the interpolation is done only along the boundary of the refined flagged region and is followed by the solution of the conservation equations in the flagged region. The two operations together represent the prolongation step. For the restriction operation, special restriction operators from the fine to the coarse grid are derived in the overlapping flagged regions. Thus in this multigrid approach the solution V^0 in the outer grid Ω^0 is obtained first, and the outer grid solution is prolongated to the embedded refined grid Ω^1 by interpolating the boundary conditions for the refined grid from the outer grid solution and then obtaining the refined grid solution V^1. The solution is then prolongated to the next level of the refined mesh, Ω^2. This process is continued to the finest mesh refinement level desired (Ω^n). Once the solution in the finest mesh has been obtained, the solution process steps back to its outer mesh Ω^{n-1} and recalculates the solution with restriction or correction terms added to the outer grid equations in the fine grid region Ω^n. These correction terms are such that the resulting corrected solution V^{n-1} is equal to V^n in the overlap region Ω^n. This process is continued until the outermost grid Ω^0 is reached. This completes one prolongation sweep and one restriction sweep. The second prolongation sweep is then initiated, and at any refinement level i, since the outer solution V^{i-1} is more accurate in view of the correction terms added in the previous restriction sweep, the boundary conditions interpolated along the boundaries of Ω^i from V^{i-1} will be more accurate and therefore the solution V^i will be better than the solution in the previous prolongation sweep. These sweeps can be continued to the desired levels of accuracy.

In the present paper most of the results have been obtained with either one or two levels of refinement and four or five prolongation and restriction sweeps. This has been found to be sufficient for the problems studied in this paper.

The coarse to fine grid information transfer in the prolongation operation has already been described in the previous section and consists of a suitable method for interpolating the boundary conditions. As noted earlier, conservative mass flux interpolation is very important for successful calculations. Attention is now turned to the restriction or correction operation that applies the fine grid solution V^i to the coarse grid equations in Ω^i such that the resulting coarse grid solution V^{i-1} in Ω^i is equal to V^i.

In the discussion below the correction equations for the pressure and momentum are derived. The coarse or outer grid terms are identified by a superscript $i-1$ and the fine grid terms by a superscript i.

The correction equations for pressure are derived by requiring that the mass fluxes across a control volume face based on both the fine and coarse grid solutions must be equal. In mathematical terms this can be written as

$$G_1^i = G_1^{i-1}, \qquad G_2^i = G_2^{i-1}. \tag{32}$$

Using equations (17) and (18), the above equations can be written as

$$\hat{G}_1^i + (B_1^i y_\eta - B_2^i x_\eta) p_\xi^i + (C_1^i y_\eta - C_2^i x_\eta) p_\eta^i - \hat{G}_1^{i-1} = (B_1^{i-1} y_\eta - B_2^{i-1} x_\eta) p_\xi^{i-1} + (C_1^{i-1} y_\eta - C_2^{i-1} x_\eta) p_\eta^{i-1}, \tag{33}$$

$$\hat{G}_2^i+(C_2^i x_\xi - C_1^i y_\xi)p_\eta^i+(B_2^i x_\xi - B_1^i y_\xi)p_\xi^i-\hat{G}_2^{i-1}=(C_2^{i-1}x_\xi - C_1^{i-1}y_\xi)p_\eta^{i-1} + (B_2^{i-1}x_\xi - B_1^{i-1}y_\xi)p_\xi^{i-1}. \tag{34}$$

By adding the above two equations, integrating over a typical control volume and using central differences, the following corrector equation for the pressure is obtained:

$$a_P^{i-1}p_P^{i-1}=a_E^{i-1}p_E^{i-1}+a_W^{i-1}p_W^{i-1}+a_N^{i-1}p_N^{i-1}+a_S^{i-1}p_S^{i-1}+b^{i-1} \quad \text{in } \Omega^i, \tag{35}$$

while the standard pressure correction equation (22) applies in the non-overlapping region $\Omega^{i-1}-\Omega^i$. In equations (33) and (34) the coefficients contain the discretized forms of $B_1 y_\eta - B_2 x_\eta$, $C_1 y_\eta - C_2 x_\eta$, $C_2 x_\xi - C_1 y_\xi$ and $B_2 x_\xi - B_1 y_\xi$. The source term b^{i-1} is given by the sum of the left-hand sides of equations (33) and (34) and contains information transferred from the fine grid.

The above equation can be derived in another way. If an operator L is defined as

$$L=(B_1 y_\eta - B_2 x_\eta)\partial_\xi+(C_1 y_\eta - C_2 x_\eta)\partial_\eta+(C_2 x_\xi - C_1 y_\xi)\partial_\eta+(B_2 x_\xi - B_1 y_\xi)\partial_\xi, \tag{36}$$

where ∂_ξ and ∂_η denote $\partial/\partial\xi$ and $\partial/\partial\eta$ respectively, since in the region Ω^i it is desired that $p^{i-1}=p^i$, the corrector equation that will yield this is

$$L^{i-1}p^{i-1}=L^{i-1}p^i \quad \text{in } \Omega^i. \tag{37}$$

It can be easily shown that equation (37) leads to equation (35), while in the remaining region the equation

$$L^{i-1}p^{i-1}=b_P \quad \text{in } \Omega^{i-1}-\Omega^i, \tag{38}$$

with b_P given by equation (25), can be shown to be equivalent to equation (22).

The derivations of the corrector equations for the x- and y-momentum are similar to those of a scalar variable,[17] with the exception of the pressure gradient term. However, by suitable manipulation these terms can be added to the total flux. Since

$$-p_x=(-y_\eta p_\xi + y_\xi p_\eta)/J=[-(y_\eta p)_\xi+(y_\xi p)_\eta]/J, \tag{39}$$

the x-momentum equation can be rewritten as

$$[\rho G_1 u-(\Gamma/J)(\alpha u_\xi-\beta u_\eta)+y_\eta p]_\xi+[\rho G_2 u-(\Gamma/J)(\gamma u_\eta-\beta u_\xi)-y_\xi p]_\eta=0. \tag{40}$$

Conservation of momentum flux on both the coarse and fine grids implies that the momentum fluxes based on the two solutions can be equated to each other, leading to

$$[\rho G_1 u-(\Gamma/J)(\alpha u_\xi-\beta u_\eta)+y_\eta p]_\xi^{i-1}+[\rho G_2 u-(\Gamma/J)(\gamma u_\eta-\beta u_\xi)-y_\xi p]_\eta^{i-1} = [\rho G_1 u-(\Gamma/J)(\alpha u_\xi-\beta u_\eta)+y_\eta p]_\xi^{i}+[\rho G_2 u-(\Gamma/J)(\gamma u_\eta-\beta u_\xi)-y_\xi p]_\eta^{i}. \tag{41}$$

Since $p^{i-1}=p^i$ (equation (37)), the pressure terms can be dropped and the remaining discretization done as for equation (12), resulting in a correction equation for the x-momentum in Ω^i. A similar correction equation for the y-momentum can be obtained. These correction equations are solved in Ω^i, while the conventional discretized momentum equations (equations (14) and (16)) are solved in $\Omega^{i-1}-\Omega^i$.

As for the pressure equation, the corrector equations for momentum can also be derived by using the equation $L^{i-1}u^{i-1}=L^{i-1}u^i$ in Ω^i, where L^{i-1} is the appropriate discretized momentum operator.

Overall algorithm

The overall algorithm between two grid levels 0 and 1 can be summarized as follows.

1. Generate a preliminary grid in Ω^0 in the domain by solving equations (1) and (2). Solve the conservation equations $L^0\phi^0=f$ on this grid.
2. Calculate the normalized error estimate $\tilde{W}_\phi$ at each grid point; where $\tilde{W}_\phi$ is greater than a critical value, the grid point is flagged. Contiguous flagged points form a cluster and define a flagged region Ω^1.
3. For each flagged region Ω^1 identify the new computational boundaries $\eta'_{\min}$, $\eta'_{\max}$, $\xi'_{\min}$ and $\xi'_{\max}$. In each flagged region calculate the local values of P^* and Q^* using equations (8) and (9). Then compute the new equidistributed mesh in each flagged region by solving equations (1) and (2).

Prolongation step

4. Along the boundaries of each flagged region Ω^1 interpolate the outer coarse grid solution ϕ^0 to the inner fine grid boundary points. The new boundary values ϕ_B^1 can be interpreted as $I_0^1\phi^0$, where I_0^1 is the prolongation operator in the multigrid terminology.
5. With the new interpolated boundary values ϕ_B^1 solve the conservation equation $L^1\phi^1=f$ on the refined grid in each flagged region Ω^1.

Restriction step

6. Derive corrected or restricted conservation equations in the overlap region (equation (35) or (37) and similar equations for other variables). The complete set of corrected or restricted equations has the form

$$L^0\phi^0=L^0\phi^1 \quad \text{in } \Omega^1, \tag{42a}$$

$$L^0\phi^0=f \quad \text{in } \Omega^0-\Omega^1. \tag{42b}$$

The right-hand side of equation (42a) can be interpreted as $f^0+I_1^0(f^1-L^1\phi^1)$, where I_1^0 is the restriction operator. Solve equations (42a) and (42b) for all variables to obtain the updated outer grid solution.

7. Return to step 4 and repeat steps 4–6 until the desired level of accuracy is obtained.

The above algorithm can be easily extended to larger numbers of grid levels.

RESULTS AND DISCUSSION

Four test problems are studied. The first deals with driven flow in a square cavity. This test problem is commonly solved in order to test the performance of a numerical method. In the second problem laminar isothermal flow past a backward-facing step is considered. The third and fourth test problems also deal with flow over a backward-facing step, but with heat transfer effects included. For all the problems the adaptive grid solution initiated on an $N\times N$ grid is compared with two fixed grid solutions, one on an $N\times N$ grid and other on an $(N+n)\times(N+n)$ fixed grid that requires the same CPU effort as the adaptive grid calculation. The second comparison is more meaningful, since for the same computational effort it provides a comparison of the solution accuracy between the adaptive grid procedure and the fixed grid procedure. In all the cases considered the constants α_1 and α_2 in equation (27) are set to unity while α_3 and α_4 are set to zero.

Driven flow in a square cavity

The physical situation is shown in Figure 4. If L denotes the cavity dimensions and u_S is the velocity of the upper plate, the following dimensionless variables can be defined:

$$X=x/L, \qquad Y=y/L, \qquad U=u/u_S, \qquad V=v/u_S, \qquad P=p/\rho u_S^2. \tag{43}$$

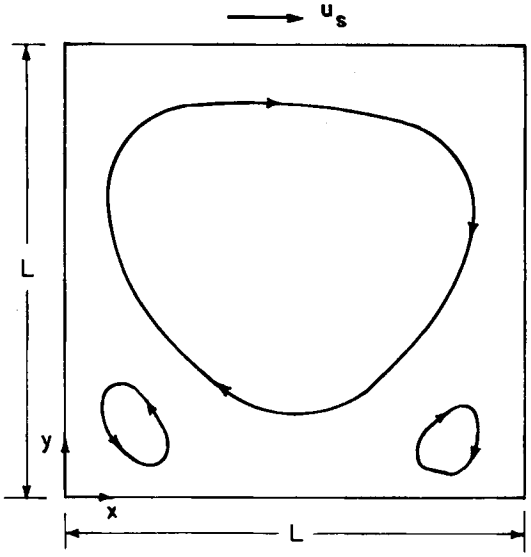

Figure 4. Schematic diagram of driven cavity flow

The dimensionless mass and momentum conservation equations are then given by

$$\nabla \cdot \mathbf{U} = 0, \tag{44}$$

$$\mathbf{U} \cdot \nabla U = -P_X + \nabla^2 U/Re, \tag{45}$$

$$\mathbf{U} \cdot \nabla V = -P_Y + \nabla^2 V/Re, \tag{46}$$

where $Re = \rho u_S L/\mu$ is the Reynolds number. A value of $Re = 100$ is used in this paper. The boundary conditions are zero velocities along all surfaces except the top plate, where $U = 1$ and $V = 0$.

Calculations are initiated on an 11×11 grid and are performed for two values of $\tilde{W}_\phi$. For $\tilde{W}_\phi = 0{\cdot}3$ the original mesh (broken lines) and the refined mesh after the first level of refinement (solid lines) are shown in Figure 5. It can be seen that the flagged region is located in the upper half of the domain where higher gradients are expected owing to the moving wall. The flagged region is nearly rectangular and the refined mesh is nearly orthogonal. Since the gradients and curvatures do not change rapidly in this situation, the refined equidistributed grid is not highly non-uniform. Figure 6 shows the refined mesh for $\tilde{W}_\phi = 0{\cdot}4$. Fewer grid points are flagged owing to the higher error level specified for flagging a point. The flagged region is seen to be a non-rectangular, irregularly shaped domain, and it is for this reason that the case $\tilde{W}_\phi = 0{\cdot}4$ is included here in addition to the $\tilde{W}_\phi = 0{\cdot}3$ case. The $\xi'_{\min}$ and $\xi'_{\max}$ boundaries have been chosen to be AB and CD respectively, and correspondingly AC and BD are the $\eta'_{\min}$ and $\eta'_{\max}$ boundaries.

Figure 7 shows the U- and V-velocities along the vertical centreline ($x/L = 0{\cdot}5$) and Figure 8 shows the velocity profiles along the horizontal centreline ($y/L = 0{\cdot}5$). The 'correct' solution for purposes of comparison is taken to be that of Burgraff.[27] In addition to Burgraff's solution, three other solutions are shown. These are the fixed grid solution on an 11×11 grid, the adaptive grid solution initiated on an 11×11 grid and a 20×20 fixed grid solution that requires the same CPU effort as the adaptive grid. The results clearly indicate that the adaptive grid solution initiated on an 11×11 grid is superior to the 11×11 fixed grid solution and comparable to the 20×20 fixed grid solution. What is particularly noteworthy is that the adaptive grid solution in the lower half of the domain (the non-flagged region) is superior to the fixed grid (11×11) solution even if the

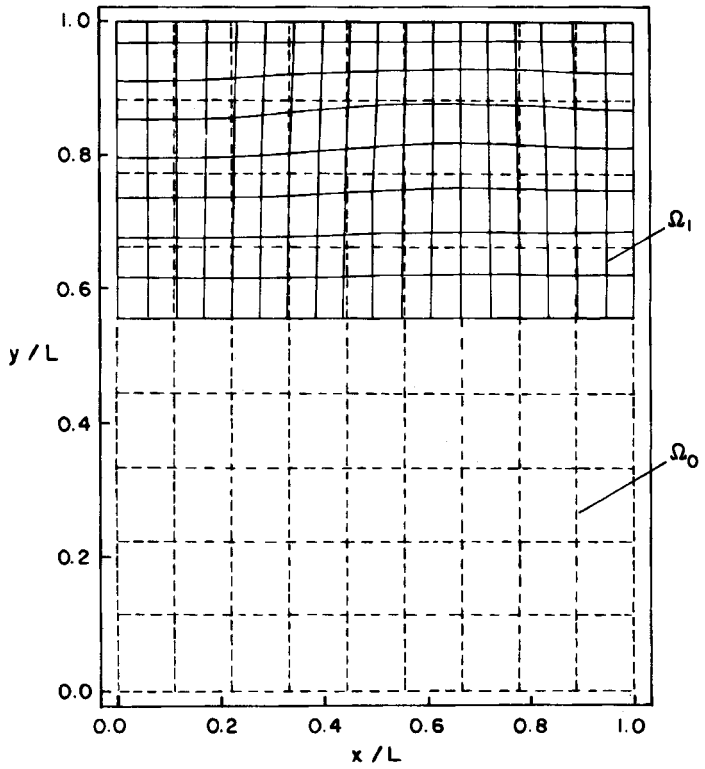

Figure 5. Mesh for driven cavity flow; $Re=100$, $\tilde{W}_\phi=0{\cdot}3$. Note: broken lines (– – –) represent initial coarse grid (Ω_0); solid lines (——) represent adapted grid after first level of refinement (Ω_1)

two grids in this region are the same. This demonstrates the effectiveness of the multigrid procedure in reducing errors in the outer grid.

Figure 9 shows the velocity profiles along the vertical centreline for $\tilde{W}_\phi=0{\cdot}4$ and exhibits similar features to those observed in Figures 7 and 8. In making this comparison it is important to stress that adaptive grid results are being obtained on a curvilinear non-orthogonal mesh and are being compared with fixed grid results obtained on a Cartesian orthogonal mesh. Calculations on a non-orthogonal mesh are intrinsically less accurate because of the discretization approximations required for the metric tensors such as α, β, γ and J appearing in equation (12). Despite this, the adaptive grid solution is superior, and if it were compared with a non-orthogonal curvilinear fixed grid calculation, it is expected to be even better.

Isothermal flow over a backward-facing step

Separated flows over backward-facing steps arise in many practical situations such as flow in electronic passages, flow in combustors, flow in ribbed a heat exchangers, etc. Isothermal laminar flow over a backward-facing step is considered here, as the second test problem, and results are compared with the experimental data of Armaly *et al.*[28]

The schematic diagram of the physical situation is shown in Figure 10. The step height s and the channel height h are chosen to match the values in Reference 28. The computational domain

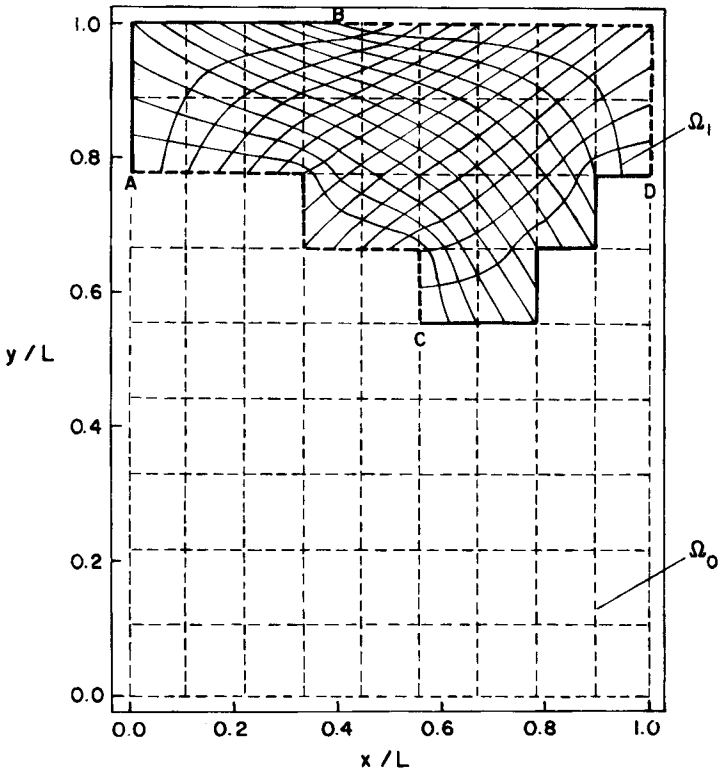

Figure 6. Mesh for driven cavity flow; $Re = 100$, $\tilde{W}_\phi = 0{\cdot}4$. See note in Figure 5

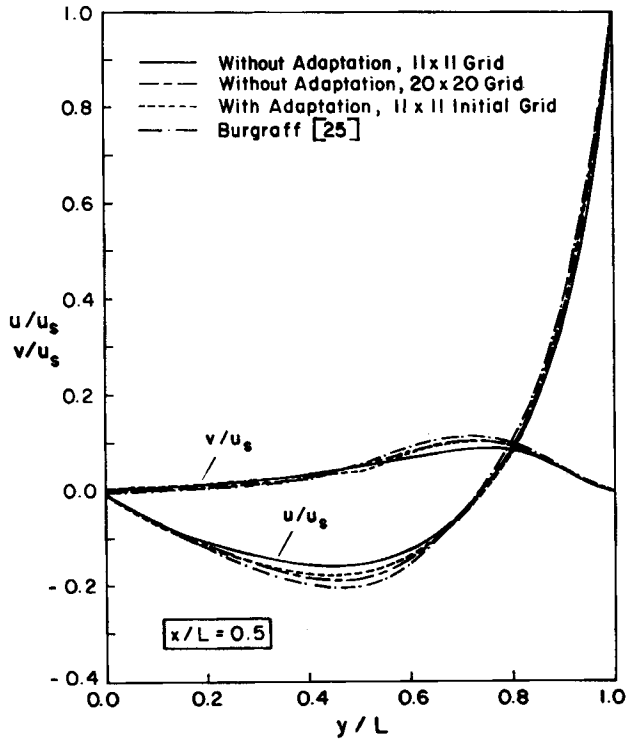

Figure 7. Vertical centreline ($x/L = 0{\cdot}5$) velocity profile; $Re = 100$, $\tilde{W}_\phi = 0{\cdot}3$

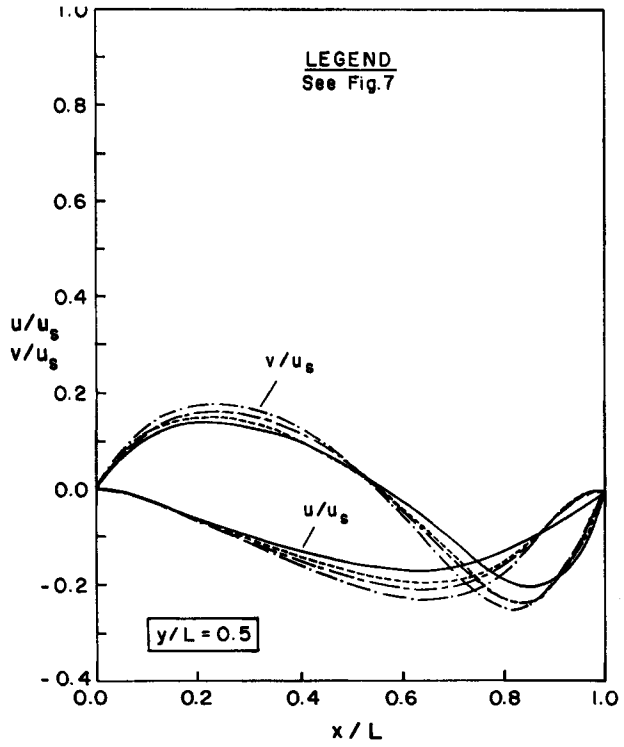

Figure 8. Horizontal centreline ($y/L=0{\cdot}5$) velocity profile; $Re=100$, $\tilde{W}_\phi=0{\cdot}3$

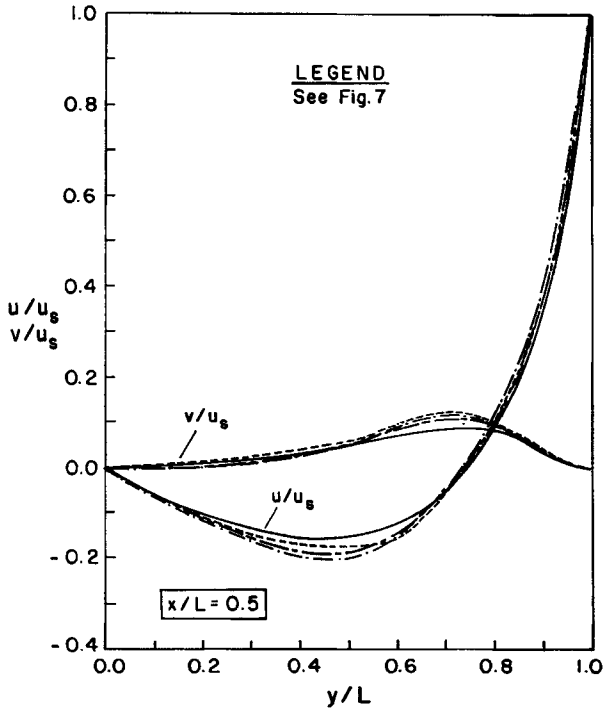

Figure 9. Vertical centreline ($x/L=0{\cdot}5$) velocity profile; $Re=100$, $\tilde{W}_\phi=0{\cdot}4$

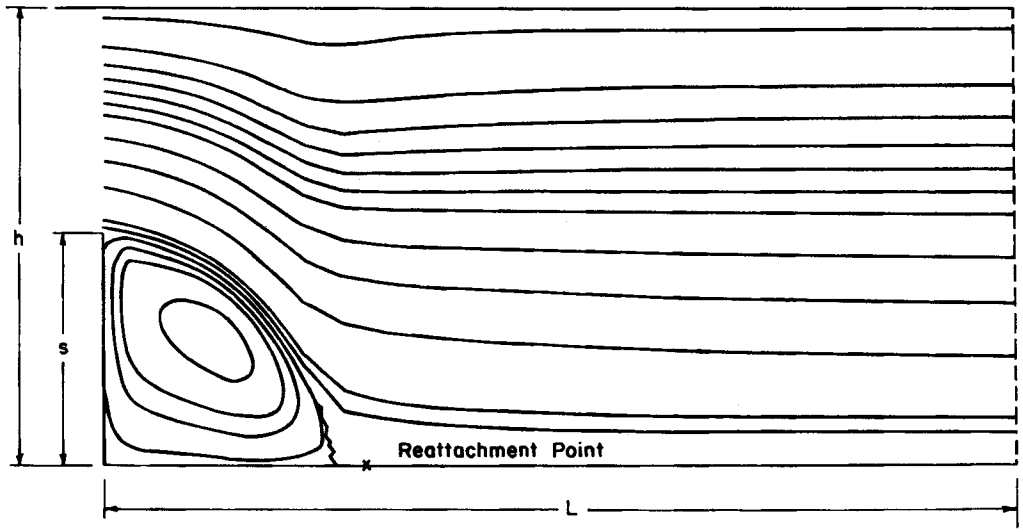

Figure 10. Flow past a backward-facing step

in the streamwise direction is 30s. The governing equations can be written as

$$\nabla\cdot(\rho\mathbf{u})=0, \tag{47}$$

$$\nabla\cdot(\rho\mathbf{u}u)=p_x+\mu\nabla^2u, \tag{48}$$

$$\nabla\cdot(\rho\mathbf{u}v)=p_y+\mu\nabla^2v, \tag{49}$$

where u and v are the x- and y-velocity components and p is the pressure. The boundary conditions used at the inlet are a parabolic u-profile with a Reynolds number based on a hydraulic diameter of $2h$ equal to 389. At all the solid surfaces u and v are assumed to be zero and at the outflow all streamwise gradients are assumed to be zero.

The cross-stream u-velocity profiles are plotted in Figures 11–13 at different x/s locations. The adaptive grid solution (initiated on an 11×11 grid) is compared with solutions obtained using the conventional grid scheme (on 11×11, 30×30 and 45×45 grids) and with experimental results. The 30×30 fixed grid solution takes the same CPU effort as the adaptive grid solution. The 45×45 grid solution was obtained so that it will serve as a standard for comparison in addition to the experimental data. From these results it is clear that the adaptive grid solution is at least as good as, and is generally better than, the conventional grid solution obtained on a 30×30 grid that takes the same computer time. The improvement is not confined to the refined regions only (the refined region is not shown but is similar to that shown in Figure 14). In fact, profiles in non-refined regions show noticeable improvement. This has already been demonstrated in the first test problem.

Flow over a backward-facing step—heating along the entire bottom surface

This case is similar to the one considered previously but with the entire lower surface heated. The computational domain extends from 3s upstream of the step to 20s downstream of the step. In addition to equations (47)–(49), the energy equation is solved,

$$\nabla\cdot(\rho\mathbf{u}\theta)=(\mu/Pr)\nabla^2\theta, \tag{50}$$

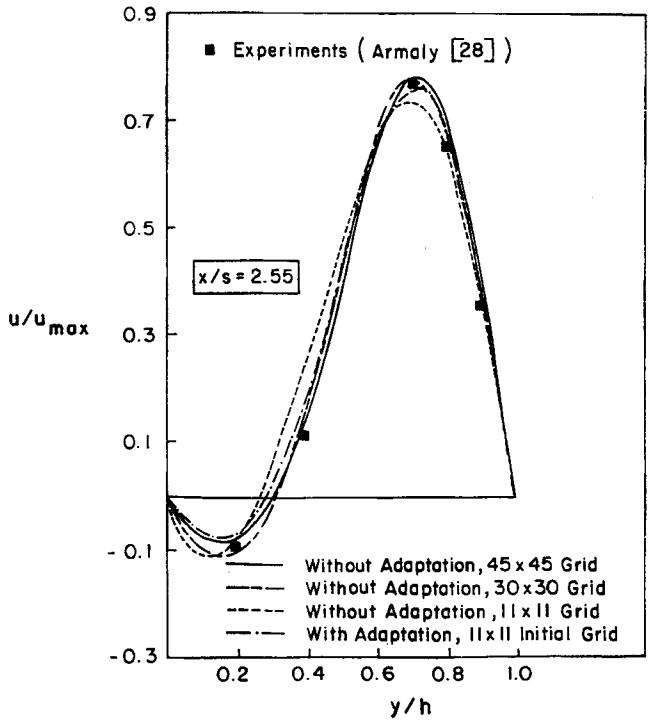

Figure 11. Streamwise velocity profile for isothermal flow past a backward-facing step; $x/s=2{\cdot}55$

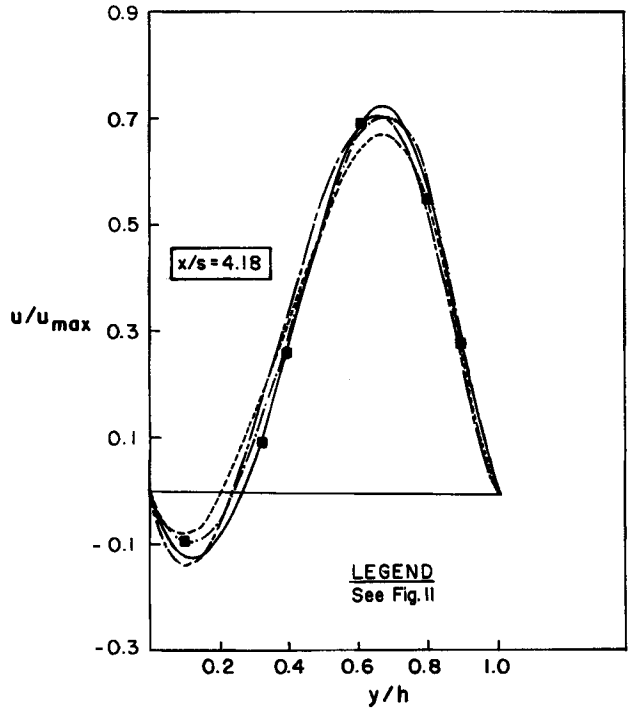

Figure 12. Streamwise velocity profile for isothermal flow past a backward-facing step; $x/s=4{\cdot}18$

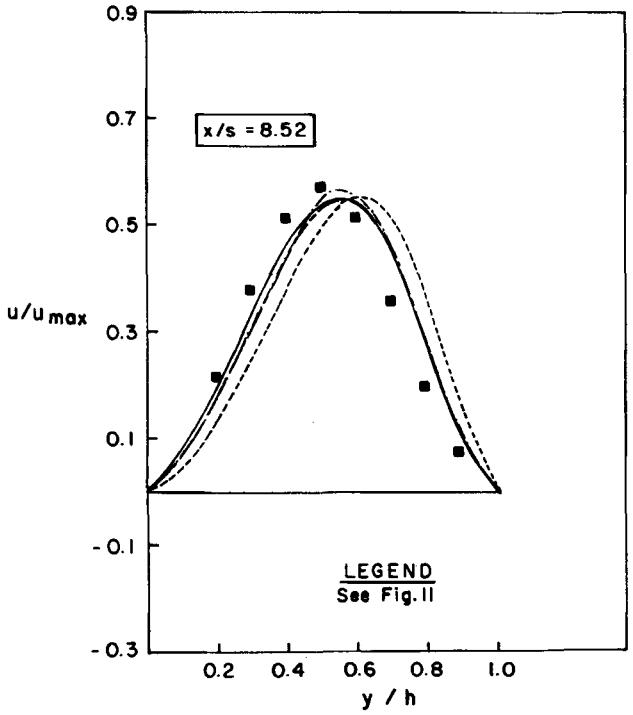

Figure 13. Streamwise velocity profile for isothermal flow past a backward-facing step; $x/s=8{\cdot}52$

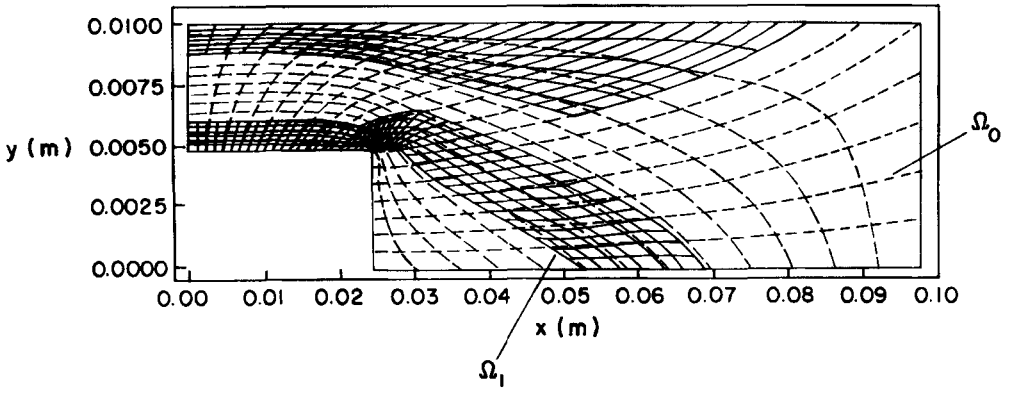

Figure 14. Initial mesh (Ω_0, broken lines) and adapted mesh after one level of refinement (Ω_1, solid lines) for flow past a step

where $\theta=(T-T_0)/(T_1-T_0)$ is the dimensionless temperature and $Pr=\mu C_p/k$ is the Prandtl number. The temperature of the incoming flow is T_0 while the temperature of the bottom surface is T_1. The velocity boundary conditions are as before while the temperature boundary conditions are

$$\theta=1 \begin{cases} \text{at} \quad y=s \quad \text{for} \quad 0\leqslant x<3s, \\ \text{at} \quad y=0 \quad \text{for} \quad 3s\leqslant x\leqslant 20s, \\ \text{at} \quad x=3s \quad \text{for} \quad 0\leqslant y\leqslant s, \end{cases} \tag{51}$$

$$\partial\theta/\partial y=0 \quad \text{at } y=h \quad \text{for all } x,$$
$$\partial\theta/\partial y=0 \quad \text{at } x=20s \quad \text{for all } y. \tag{52}$$

At the inlet a parabolic profile for u and a cubic profile for θ, with $\theta=1$ at $y=s$ and $\theta=0$ at $y=h$, are specified. For purposes of comparison a very fine grid solution on an 80×70 fixed grid is obtained.

The initial grid (20×15) and the refined grid in the flagged regions are shown in Figure 14. It can be seen that the boundary layer region along the upper surface and the shear layer region in the lower part are flagged since these represent the high-gradient and high-curvature regions. The flagged region is seen to be non-rectangular in the ξ–η space and the resulting meshes in the flagged regions are error-equidistributed. As noted earlier, these features of the present method represent significant advantages over other available methods.

The streamwise velocity profile approximately one step height downstream of the step ($x=0{\cdot}03$ m) is shown in Figure 15. The recirculation region is characterized by negative velocities. The adaptive grid solution initiated on a 20×15 grid is compared with a 20×15 fixed grid and a 35×35 fixed grid solution, the latter solution taking the same CPU effort as the adaptive grid calculation. It can clearly be seen that the adaptive grid solution is considerably superior to both the fixed grid solutions. This is clearly true in the recirculation region, even if this region is not flagged, and is because of the multigrid calculations that successively improve the outer grid solution. The velocity profile just downstream of the reattachment, at x/s approximately equal to $5{\cdot}5$ ($x=0{\cdot}05$ m), is shown in Figure 16 and again clearly indicates the higher solution accuracy of the adaptive grid solution compared to the fixed grid solution requiring the same CPU effort. The temperature profile at the same location ($x=0{\cdot}05$ m) is shown in Figure 17. Over most of the

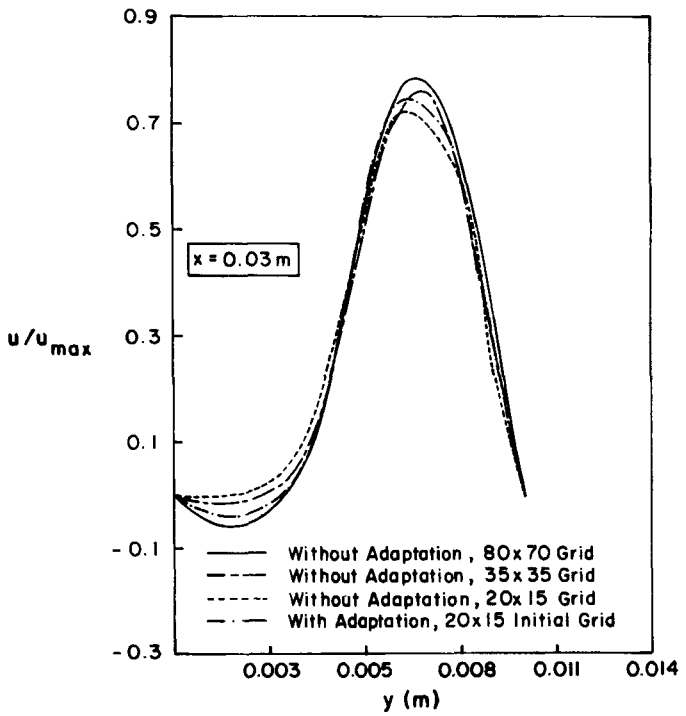

Figure 15. Streamwise velocity profile; $x=0{\cdot}03$ m. Heating along both step and reattachment surface

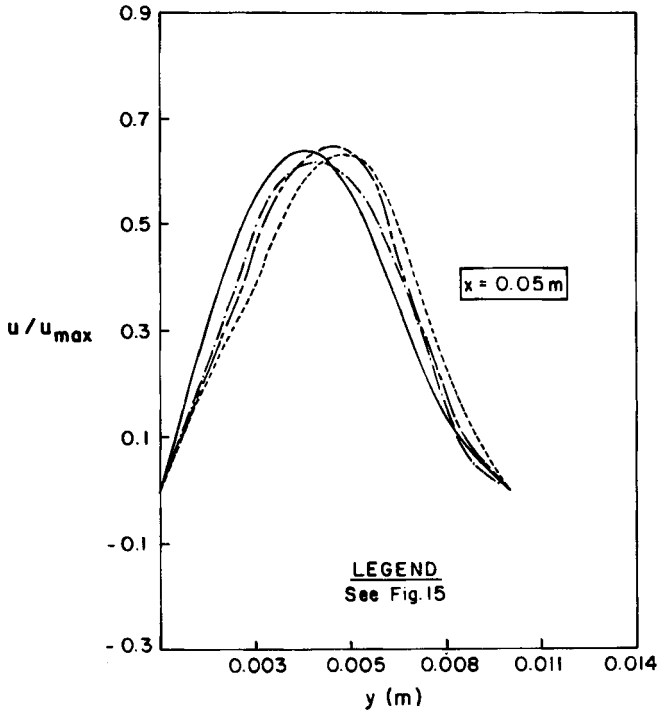

Figure 16. Streamwise velocity profile; $x = 0{\cdot}05$ m. Heating along both step and reattachment surface

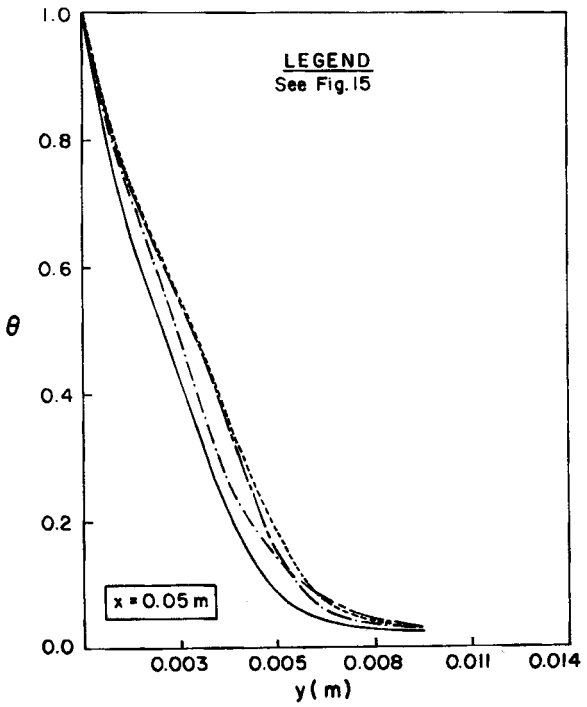

Figure 17. Temperature profile; $x = 0{\cdot}05$ m. Heating along both step and reattachment surface

cross-stream domain the adaptive grid solution is considerably superior to both the 35×35 and 20×15 fixed grid solutions.

Flow over a backward-facing step—heating downstream of the step

This case is identical to the one just considered, with the only exception that the lower surface is heated only downstream of the step. Thus the only change is that the boundary condition given by equation (51) is replaced by

$$\theta = 0 \quad \begin{cases} \text{at } y = s & \text{for } 0 \leqslant x < 3s, \\ \text{at } x = 3s & \text{for } 0 \leqslant y \leqslant s, \end{cases}$$

$$\theta = 1 \quad \text{at } y = 0 \quad \text{for } 3s \leqslant x \leqslant 20s. \tag{53}$$

This case is included here to specifically show that feature of the adaptive grid method which permits it to automatically select, at a point, the dependent variable on which to base the calculation of the weighting function. Thus the grid is optimized for all dependent variables.

The difference between this case and the previous one is that the reattaching flow will be colder in the present case since the step is not heated. Thus higher temperature gradients in the vicinity of the reattachment are expected compared to the previous case, and as a consequence it is expected that for the present case the flagged region will be larger near the reattachment location. This is confirmed by comparing Figure 18 with Figure 14. Near the reattachment point the weighting functions are controlled by temperature while near the top of the step, where the flow separates, the weighting function is based on velocities. Thus the needs of all dependent variables are accommodated.

Figures 19 and 20 show the temperature profiles at $x = 0{\cdot}05$ m and 0·07 m respectively. At both streamwise locations the adaptive grid solutions initiated on a 20×15 grid are considerably better than the 20×15 fixed grid solution and the 35×35 fixed grid solution requiring the same CPU effort. This is true in both the flagged and non-flagged regions.

Multiple levels of grid refinement

To illustrate the present procedure with multiple levels of grid refinement, the driven cavity problem is again chosen. Two levels of grid refinement were performed with a number of prolongation and restriction sweeps between each level. The three grids are shown in Figure 21

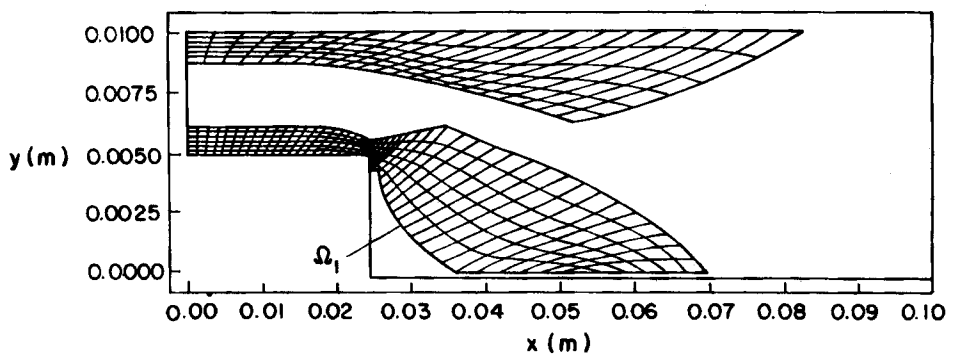

Figure 18. Adapted mesh after one refinement (Ω_1) for heating only along reattachment surface

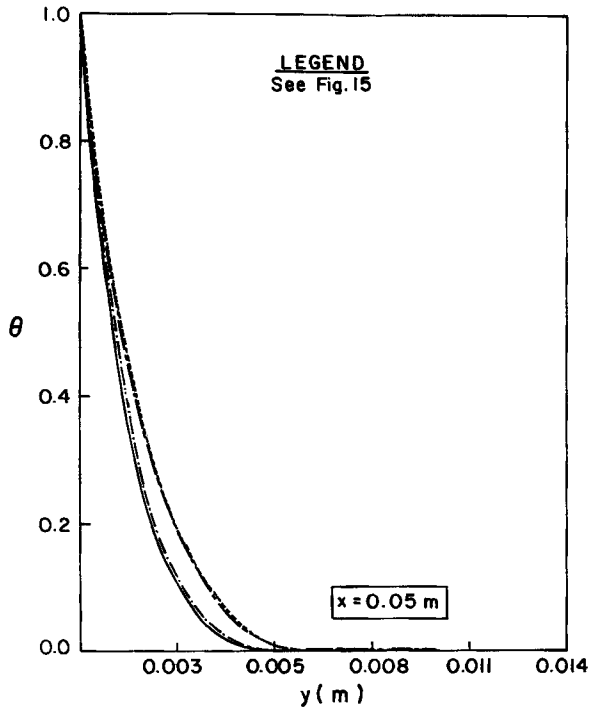

Figure 19. Temperature profile; $x = 0{\cdot}05$ m. Heating only along reattachment surface

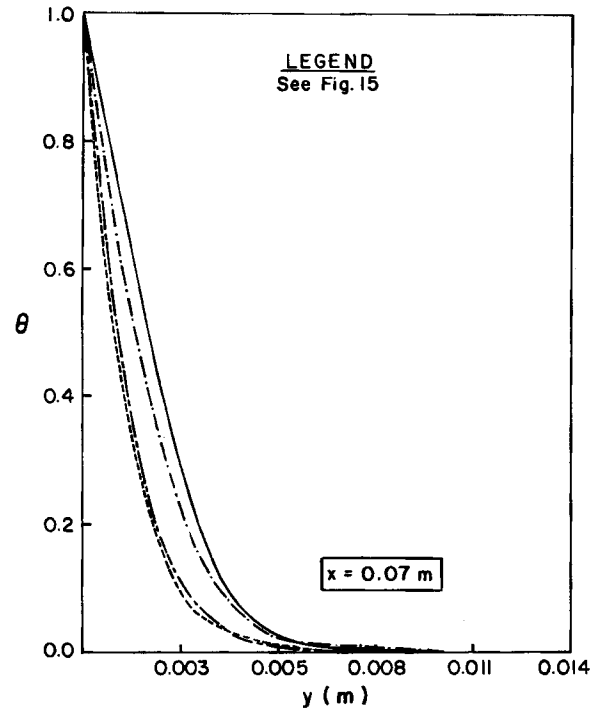

Figure 20. Temperature profile; $x = 0{\cdot}07$ m. Heating only along reattachment surface

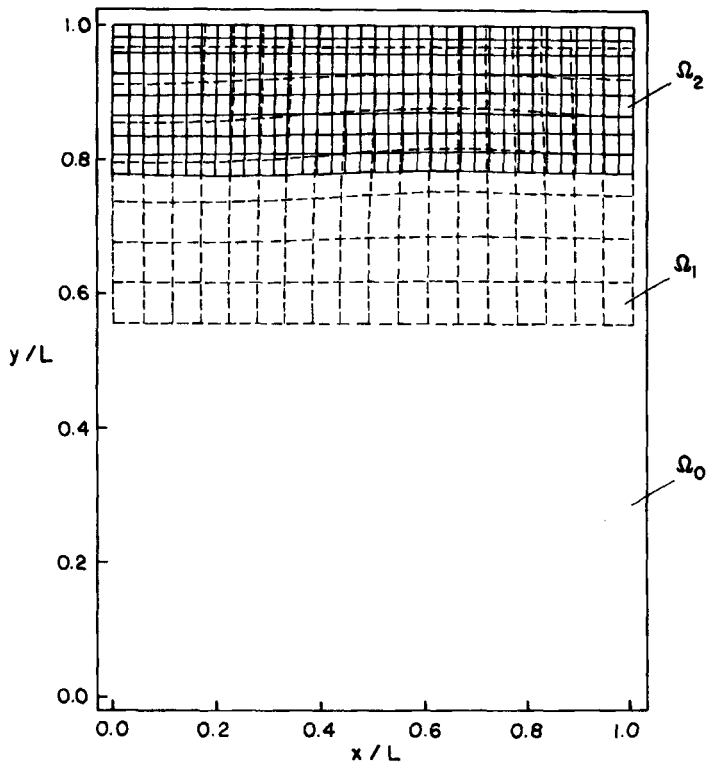

Figure 21. Mesh in driven cavity: initial mesh (Ω_0), adapted mesh after first refinement (Ω_1), adapted mesh after second refinement (Ω_2); $Re = 100$

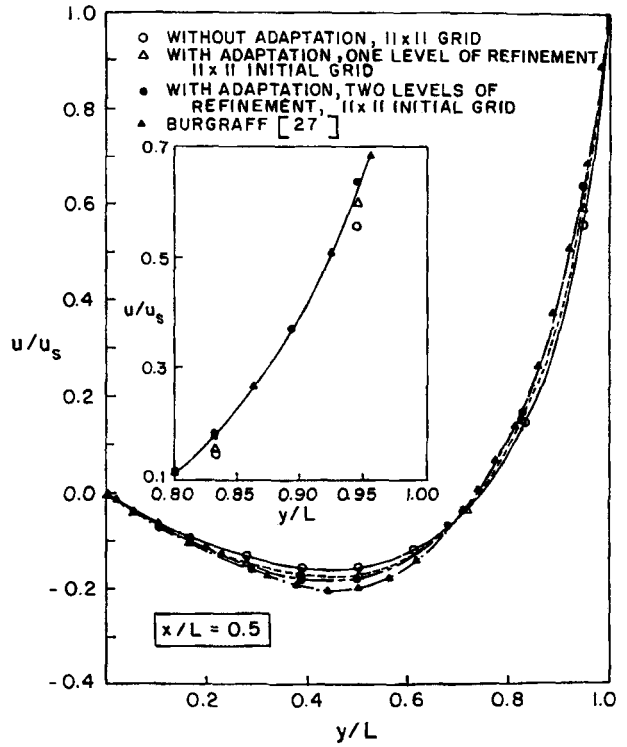

Figure 22. Vertical centreline velocity profile; $x/L = 0{\cdot}5$, $Re = 100$

and denoted as Ω^0, Ω^1 and Ω^2. As expected, at each level it is the upper part of the domain, near the moving wall, that is flagged.

The vertical centreline u-velocity profile is shown in Figure 22. Results shown include those of Burgraff,[27] those obtained on an 11×11 fixed grid and two sets of adaptive grid solutions initiated on an 11×11 grid. The first set is based on one level of refinement (results shown earlier) and the second set on two levels of refinement. These results show improvement in the calculations with each successive level of refinement. The inset in the figure shows that the vertical centreline profile in Ω^2 (the second refinement level) is clearly superior to the initial solution and the solution after the first refinement.

CONCLUDING REMARKS

A powerful solution-adaptive grid procedure is developed for flow problems. The procedure adapts only in locally important regions that are flagged from an initial solution, but grid refinement in these flagged regions is based on an error equidistribution law. Each of the flagged regions in the ξ–η computational domain can have an arbitrary shape. The solution accuracy in both the outer grid and the grid in the flagged regions is successively improved using a multigrid-type procedure. The resulting solution-adaptive grid procedure is applied to a number of test problems and the improvement in results obtained is clearly demonstrated by comparing with fixed grid calculations that take the same effort.

APPENDIX: LIST OF SYMBOLS

a_P, a_E, a_W, a_N, a_S	coefficients in finite difference equation
b_{no}	source term due to non-orthogonality in finite difference equation
b_0	source term in finite difference equation
e	east face of a control volume, Figure 1
E	east neighbour of grid point P, Figure 1
f_1	constant for a given curve in the equidistribution law
G_1, G_2	refer to contravariant velocity components
J	Jacobian
L	differential operator
n	north face of a control volume, Figure 1
N	north neighbour of grid point P, Figure 1
p	pressure
P	grid point, Figure 1
P^*	control function
Q^*	control function
s	south face of a control volume, Figure 1
s	distance along an arc
S	south neighbour of grid point P, Figure 1
T	temperature
u, U	dimensional and dimensionless velocity in x-direction
v, V	dimensional and dimensionless velocity in y-direction
V^i	adaptive grid solution at the ith refinement
w	west face of control volume, Figure 1
W	west neighbour of grid point P
W	weighting function

$\bar{W}$	normalized weighting function
x	Cartesian co-ordinate
y	Cartesian co-ordinate

Greek symbols

α	geometric parameter
$\alpha_1, \alpha_2, \alpha_3, \alpha_4$	constants
β	geometric parameter
$\delta\xi, \delta\eta$	distances between grid points in the computational plane
η	computational co-ordinate
θ_1, θ_2	slopes of constant η and ξ curves
θ_3	angle of intersection, (θ_1, θ_2)
ρ	density
ϕ	dependent variable
$\tau_1, \tilde{\tau}$	control functions
$\psi_1, \tilde{\psi}$	control functions
ξ	computational co-ordinate
Γ	diffusion coefficient
Ω^i	flagged region at the ith refinement
$\Delta\xi, \Delta\eta$	distances between interfaces in the computational plane

Subscripts

e, w, n, s	refer to control volume faces
P, E, W, N, S	refer to grid points
x, y, η, ξ	indicate differentiation with respect to x, y, η and ξ respectively
max	maximum value
min	minimum value

REFERENCES

1. J. F. Thompson, 'A survey of dynamically-adaptive grids in the numerical solution of partial differential equations', *Appl. Numer. Math.*, **1**, 3–30 (1985).
2. S. Acharya, 'A calculation procedure for parabolic flows with adaptive grid and spline techniques', *Ph.D. Thesis*, University of Minnesota, 1982.
3. H. A. Dwyer, R. J. Kee and B. R. Sanders, 'Adaptive grid method for problems in fluid mechanics and heat transfer', *AIAA J.*, **8**, 1205–1212 (1980).
4. H. A. Dwyer, 'Grid adaptation for problems in fluid dynamics', *AIAA J.*, **22**, 1705–1712 (1984).
5. P. A. Gnoffo, 'A vectorized finite-volume, adaptive grid algorithm for Navier–Stokes calculations', in J. F. Thompson (ed.), *Numerical Grid Generation*, North-Holland, Amsterdam, 1982, pp. 819–835.
6. K. Nakahashi and G. S. Deiwert, 'A self-adaptive grid method with application to air flow', *AIAA Paper 85-1525, AIAA 7th Comp. Fluid Dynamics Conference*, Cincinnatti, OH, 1985.
7. M. M. Rai and D. A. Anderson, 'Grid evolution in time asymptotic problems', *J. Comput. Phys.*, **43**, 357–372 (1981).
8. D. A. Anderson and M. M. Rai, 'The use of solution adaptive grids in solving partial differential equations', in J. F. Thompson (ed.), *Numerical Grid Generation*, North-Holland, Amsterdam, 1982, pp. 317–338.
9. J. B. Greenberg, 'A new self adaptive grid method', *AIAA Paper 83-1934, AIAA 6th Comp. Fluid Dynamics Conference*, Dauvers, MA, 1983.
10. P. R. Eiseman, 'Alternating direction adaptive grid generation', *AIAA J.*, **23**, 551–560 (1985).
11. H. J. Kim and J. F. Thompson, 'Three-dimensional adaptive grid generation on a composite-block grid', *AIAA J.*, **28**, 470–477 (1990).
12. M. J. Berger and A. Jameson, 'Automatic adaptive grid refinement for the Euler equations', *AIAA J.*, **23**, 561–568 (1985).

13. R. E. Phillips and F. W. Schmidt, 'A multilevel–multigrid technique for recirculating flows', *Numer. Heat Transfer*, **8**, 573–594 (1985).
14. M. Berger and J. Oliger, 'Adaptive mesh refinement for hyperbolic partial differential equation', *J. Comput. Phys.*, **53**, 484–512 (1984).
15. W. Skamarock, J. Oliger and R. L. Street, 'Adaptive grid refinement for numerical weather production', *J. Comput. Phys.*, **80**, 27–60 (1989).
16. M. C. Thompson and J. H. Ferziger, 'An adaptive multigrid technique for the incompressible Navier–Stokes equation, *J. Comput. Phys.*, **82**, 94–121 (1989).
17. S. Acharya and F. H. Moukalled, 'An adaptive grid solution procedure for convection–diffusion problems', *J. Comput. Phys.*, **91**, 32–54 (1990).
18. S. Acharya and F. H. Moukalled, 'Improvements to incompressible flow calculation on a non-staggered curvilinear grid', *Numer. Heat Transfer*, **15**, 63–75 (1989).
19. S. V. Patankar, 'A calculation procedure for two-dimensional elliptic situations', *Numer. Heat Transfer*, **4**, 409–425 (1981).
20. F. H. Hsu, 'A curvilinear-coordinate method for momentum, heat and mass transfer in domains of irregular geometry', *Ph.D. Thesis*, University of Minnesota, 1981.
21. M. Reggio and R. Camarero, 'Numerical solution procedure for viscous incompressible flows', *Numer. Heat Transfer*, **10**, 131–146 (1986)
22. C. M. Rhie and W. L. Chow, 'Numerical study of the turbulent flow past an airfoil with trailing edge separation', *AIAA J.*, **21**, 1525–1532 (1983).
23. J. F. Thompson, F. C. Thames and C. W. Martin, 'Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies', *J. Comput. Phys.*, **15**, 299–319 (1974).
24. D. A. Anderson and J. Steinbrenner, 'Generating adaptive grids with a conventional grid scheme', *AIAA Paper 86-0427, AIAA 24th Aerospace Science Meeting*, Reno, NV, 6–9 January 1986.
25. A. Brandt, 'Multi-level adaptive solution to boundary value problems', *Math. Comput.*, **31**, 333–390 (1977).
26. S. P. Vanka, 'Block-implicit multigrid solution of Navier–Stokes equations in primitive variables', *J. Comput. Phys.*, **65**, 138–158 (1986).
27. O. R. Burgraff, 'Analytical and numerical studies of the structure of steady separated flows', *J. Fluid Mech.*, **24**, 113–151 (1966).
28. B. F. Armaly, F. Durst, J. C. F. Pereira and B. Schonung, 'Experimental and theoretical investigation of backward-facing step flow', *J. Fluid Mech.*, **127**, 473–496 (1983).