JOURNAL OF COMPUTATIONAL PHYSICS 82, 94–121 (1989)

# An Adaptive Multigrid Technique for the Incompressible Navier–Stokes Equations

M. C. THOMPSON

*CSIRO, Division of Construction and Engineering, P.O. Box 56, Highett, 3190, Victoria, Australia*

AND

J. H. FERZIGER

*Thermosciences Division, Department of Mechanical Engineering, Stanford University, Stanford, California 94305*

Received June 29, 1987; revised June 3, 1988

An automatic adaptive refinement technique has been coupled to the multigrid approach to produce an efficient and stable solution strategy for solving the steady-state incompressible Navier–Stokes equations. Solutions have been obtained for the driven cavity and flow over a backward-facing step, for Reynolds numbers up to 5000 and 800, respectively. The refinement criterion is based on the local truncation error. The solution error is monitored and automatic refinement can continue until it is reduced to a satisfactory level. For driven cavity flow at $Re = 1000$, the adaptive refinement approach reduced the computer memory and CPU time to 20 and 40% of the requirements of the "pure" multigrid method. The primitive-variable formulation of the Navier–Stokes equations is used so the method can be extended easily to three dimensions. Application to other nonlinear elliptic problems is equally possible. © 1989 Academic Press, Inc.

## 1. INTRODUCTION

In the last few years considerable progress has been made in the development of numerical methods for solving the steady-state incompressible Navier–Stokes equations. This has resulted in techniques which are both more stable and more efficient in that less computer time and/or memory are required to obtain a solution. Among the most successful approaches is the multigrid method which uses a sequence of coarser grids to accelerate the convergence. Notable contributions using the multigrid approach include articles by Ghia *et al.* [1], Fuchs and Zhao [2], and Vanka [3–5]. Other authors have employed various adaptive gridding techniques. These methods attempt to distribute the gridpoints so that the truncation error of the discretized differential equations is small everywhere. This paper describes a method which combines the multigrid method with automatic adaptive gridding to provide the basis of a stable and very efficient solution strategy.



0021-9991/89 $3.00

Ghia *et al.* [1] used the vorticity-streamfunction formulation of the Navier–Stokes (NS) equations and employed the strongly coupled implicit technique of Rubin and Khosla [6] as their smoothing operator. Very good convergence rates were achieved. For driven cavity flow, converged solutions were obtained in approximately 20 to 100 equivalent fine grid iterations as the Reynolds number (Re) was varied from 100 to 10000. They found that the multigrid procedure decreased the computational time required by a factor of four over a single grid calculation.

Fuchs and Zhao [2] obtained solutions to the incompressible NS equations in primitive-variable form with a multigrid technique. They employed an upwind finite-difference formulation with a "distributive Gauss–Seidel" (DGS) as the smoothing operator. Relaxation of various inflow velocity profiles for 3-dimensional duct flows were examined for Re up to 2000. Of interest was their use of parabolic outflow conditions and a stretched grid in the streamwise direction.

The articles by Vanka [3–5] describe a multigrid method based on the primitive-variable formulation of the NS equations. Upwind differencing was used so the solutions are only first-order accurate; however, very fine grids are used. Vanka uses "symmetrically-coupled Gauss–Seidel" (SGGS) as the basic solver (smoother). This technique attempts to satisfy continuity for each cell at each step and appears more robust than the DGS approach (Brandt and Dinar [7], Fuchs and Zhao [2]), which has difficulty when the cell face fluxes do not match well. The 2-dimensional driven cavity has been treated for $\mathrm{Re} \leqslant 5000$ [3]. Because it is based on the primitive-variable formulation (as opposed to the vorticity-streamfunction formulation), the method is easily extended to three dimensions: indeed Vanka has obtained results for the 3-dimensional driven cavity problem [4]. The method has also been extended to include a turbulence model [5]. For a typical case, the multigrid solution technique is approximately 25 times as efficient as any of the common single grid techniques.

Recently, Barcus, Scheuerer, and Peric [8] presented a multigrid method based on the finite-volume formulation of the Navier–Stokes equations and flux averaging as the restriction operator. The smoother is an uncoupled strongly implicit solver (Stone [9]) and results similar to those of Vanka were obtained. They used a correction scheme rather than the full approximation storage scheme.

Adaptive grid methods are an efficient way of finding solutions to problems which need refinement only in small localized regions of the domain. With these methods, the grid is adjusted or refined to accurately resolve small structures without placing too many grid points in regions where the variation is smooth. Many fluids dynamics problems can be treated advantageously using this technique. High Reynolds or Mach number flows generally have thin regions of substantial velocity variation such as boundary layers and shock waves. Those regions require much finer mesh spacing than the majority of the domain. For such problems, adaptive grid techniques are much more efficient than uniform fine-grid solution techniques. Adaptive techniques can be classified into two main categories: "global" and "local" methods.

In the former class, the total number of grid points is generally fixed. The mesh is adjusted so that the highest grid point densities occur in regions in which the solution has the most rapid variation. These methods are called "global" since the entire computational domain is involved in the adjustment process. The various approaches differ in the method of moving, and the criteria for distributing, the grid points.

With the local refinement techniques, grid points are added or deleted locally so that the solution is obtained to a specified accuracy. The total number of grid points generally changes. The way grid points are added or removed varies from method to method. Reviews of the various strategies can be found in articles and reports by Thompson [10], Anderson [11], Hedstrom and Rodrigue [12], and Caruso [13].

The adaptive grid approach used in this work is a local refinement technique. It closely follows the work of Caruso [13] who used local adaptive refinement to compute laminar backward-facing step flows for Reynolds numbers up to 800. The approach involves overlaying patches of fine grid in regions where the truncation (or solution) error is large. A solution is then calculated on the composite grid. The (automatic) patching and solving continues until the solution error estimate is reduced to a satisfactory level.

## 2. METHOD

In this section first- and second-order finite difference approximations to the NS equations are presented. Then the "defect" or "deferred" correction scheme is described enabling second-order accurate solutions to be obtained while approximately maintaining the convergence properties of first-order schemes. Following this, the multigrid implementation is discussed. Then, a brief review of the adaptive scheme is given. And finally, the incorporation of the adaptive strategy into a multigrid framework is described.

### 2.1. *The Primitive Equations*

The dimensionless incompressible steady-state NS equations in conservation form may be written

$$\frac{\partial(uu)}{\partial x}+\frac{\partial(uv)}{\partial y}=-\frac{\partial P}{\partial x}+\frac{1}{\mathrm{Re}}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right), \tag{2.1.1}$$

$$\frac{\partial(uv)}{\partial x}+\frac{\partial(vv)}{\partial y}=-\frac{\partial P}{\partial y}+\frac{1}{\mathrm{Re}}\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right), \tag{2.1.2}$$

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0, \tag{2.1.3}$$

where $u$ and $v$ are the $x$ and $y$ velocity components and $P$ is the pressure. (These

equations were made non-dimensional by: $\bar{x} = x/L$, $\bar{y} = y/L$, $\bar{u} = u/U$, $\bar{v} = v/U$, and $\bar{P} = P/(\rho U^2)$. Here $L$ and $U$ are typical length and velocity scales and $\mathrm{Re} = UL/\nu$ is the Reynolds number with $\nu$ the viscosity coefficient. The overbars denoting the non-dimensionalised variables have been dropped for simplicity.) These equations are discretized on a staggered grid, (see Fig. 1), using the finite volume approach of Patankar [14]. Second-order accuracy is achieved by using a "defect-correction" technique (e.g., Auzinger and Stetter [15], Stetter [16], Hemker [17]). Initially, first- and second-order schemes are described and then an outline of the defect-correction procedure is given.

## 2.2. *The Power Law Method*

This finite-volume approximation chosen for the convection and diffusion terms is due to Patankar [14, 18]. It is based on a power-law fit to an exponential solution of the 1-dimensional convection-diffusion equation. It is similar to the scheme of Allen and Southwell [19]. This *power-law* scheme acts very much like Spalding's

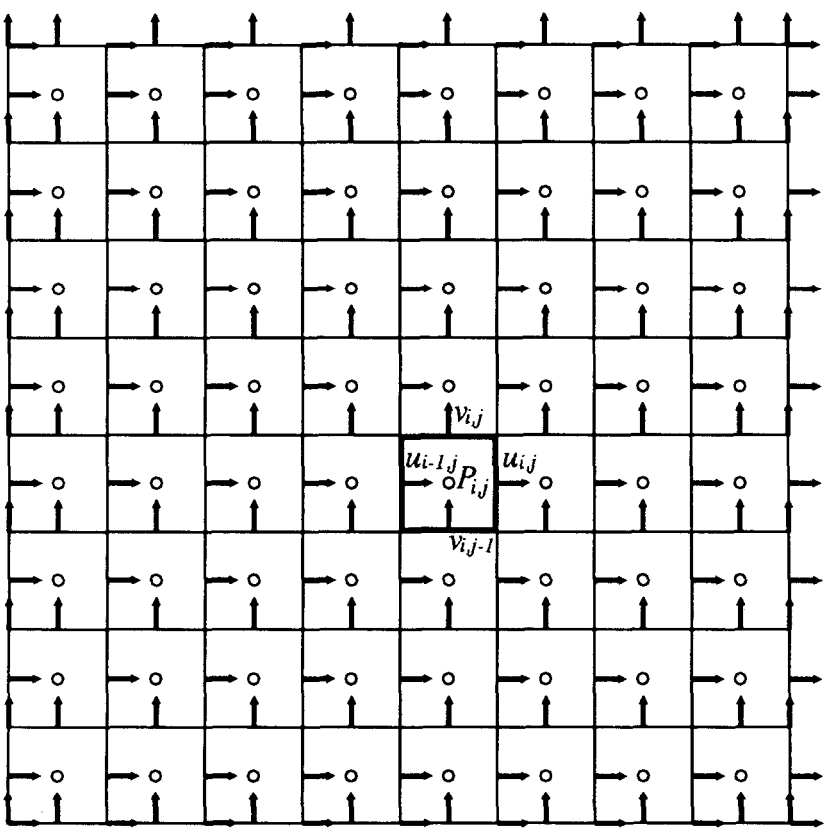


FIG. 1. Staggered grid arrangement showing the locations of $u$, $v$, and $P$. Note the irregular spacing at the boundaries of the domain.

hybrid scheme [20] except that the transition from central to upwind differencing as a function of cell Reynolds number ($\mathrm{Re}_{\Delta x} = u\,\Delta x/\nu$) is gradual. (Here $\Delta x$ denotes the cell width.) It is first-order accurate for high Reynolds numbers.

The scheme can be viewed as a finite-volume approach on a uniform grid. Its application to the $x$ momentum equation is as follows. The (north, south, east, and west) volume fluxes are computed from:

$$F_n = (v_{i+1,j} + v_{i,j})\,\Delta x/2, \tag{2.2.1a}$$

$$F_s = (v_{i+1,j-1} + v_{i,j-1})\,\Delta x/2, \tag{2.2.1b}$$

$$F_e = (u_{i+1,j} + u_{i,j})\,\Delta y/2, \tag{2.2.1c}$$

$$F_w = (u_{i,j} + u_{i-1,j})\,\Delta y/2. \tag{2.2.1d}$$

The coefficients of the discrete approximation to the momentum equations are

$$A_n = dn\,\mathscr{H}\left(\left|\frac{F_n}{dn}\right|\right) + \max(-F_n, 0), \tag{2.2.2a}$$

$$A_s = ds\,\mathscr{H}\left(\left|\frac{F_s}{ds}\right|\right) + \max(F_s, 0), \tag{2.2.2b}$$

$$A_e = de\,\mathscr{H}\left(\left|\frac{F_e}{de}\right|\right) + \max(-F_e, 0), \tag{2.2.2c}$$

$$A_w = dw\,\mathscr{H}\left(\left|\frac{F_w}{dw}\right|\right) + \max(F_w, 0), \tag{2.2.2d}$$

and

$$A_c = A_n + A_s + A_e + A_w. \tag{2.2.2e}$$

The "power-law" function $\mathscr{H}$ is defined as

$$\mathscr{H}(z) = \max(0, (1 - 0.1z)^5), \tag{2.2.3}$$

and the remaining undefined coefficients are given by

$$dn = \Delta x/(\mathrm{Re}\,\Delta y),$$

$$ds = \Delta x/(\mathrm{Re}\,\Delta y),$$

$$de = \Delta y/(\mathrm{Re}\,\Delta x),$$

$$dw = \Delta y/(\mathrm{Re}\,\Delta x).$$

With these definitions, the $x$ momentum equation can be written as

$$A_c u_{i,j} - [A_n u_{i,j+1} + A_s u_{i,j-1} + A_e u_{i+1,j} + A_w u_{i-1,j}] + \Delta y(P_{i,j} - P_{i-1,j}) = 0. \tag{2.2.4}$$

The formulation for the $y$ momentum equation is analogous and can be derived by symmetry. Finally, the continuity equation is approximated by central differences

$$(u_{i,j} - u_{i-1,j})\,\Delta y + (v_{i,j} - v_{i,j-1})\,\Delta x = 0. \tag{2.2.5}$$

## 2.3. *Central Differencing*

Second-order central-difference approximations to the $x$ and $y$ momentum equations can be written

$$\begin{aligned}
&\Delta y\left(\left(\frac{u_{i,j}+u_{i+1,j}}{2}\right)^2 - \left(\frac{u_{i-1,j}+u_{i,j}}{2}\right)^2\right)\\
&\quad + \Delta x\left(\left(\frac{u_{i,j}+u_{i,j+1}}{2}\right)\left(\frac{v_{i,j}+v_{i+1,j}}{2}\right)\right.\\
&\quad \left. - \left(\frac{u_{i,j-1}+u_{i,j}}{2}\right)\left(\frac{v_{i,j-1}+v_{i+1,j-1}}{2}\right)\right)\\
&\quad - \frac{\Delta y}{\mathrm{Re}\,\Delta x}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j})\\
&\quad - \frac{\Delta x}{\mathrm{Re}\,\Delta y}(u_{i,j+1} - 2u_{i,j} + u_{i,j-1})\\
&\quad + \Delta y(P_{i+1,j} - P_{i,j}) = 0,
\end{aligned} \tag{2.3.1}$$

$$\begin{aligned}
&\Delta y\left(\left(\frac{u_{i,j}+u_{i,j+1}}{2}\right)\left(\frac{v_{i,j}+v_{i+1,j}}{2}\right)\right.\\
&\quad \left. - \left(\frac{u_{i-1,j}+u_{i-1,j+1}}{2}\right)\left(\frac{v_{i-1,j}+v_{i,j}}{2}\right)\right)\\
&\quad + \Delta x\left(\left(\frac{v_{i,j}+v_{i,j+1}}{2}\right)^2 - \left(\frac{v_{i,j-1}+v_{i,j}}{2}\right)^2\right)\\
&\quad - \frac{\Delta y}{\mathrm{Re}\,\Delta x}(v_{i+1,j} - 2v_{i,j} + v_{i-1,j})\\
&\quad - \frac{\Delta x}{\mathrm{Re}\,\Delta y}(v_{i,j+1} - 2v_{i,j} + v_{i,j-1})\\
&\quad + \Delta x(P_{i,j+1} - P_{i,j}) = 0.
\end{aligned} \tag{2.3.2}$$

The differencing of the continuity equation is still given by Eq. (2.2.5). These equations are linearized when employed in the iteration scheme.

Special treatment is needed at the boundary cells where the staggered grid is nonuniform (see Fig. 1). For the $u$-grid the vertical spacing varies near horizontal walls.

The term $\partial^2 u/\partial y^2$, which is important near horizontal solid walls, can be estimated by

$$\frac{\partial^2 u}{\partial y^2} = \frac{2u_{0,j} - 3u_{1,j} + u_{2,j}}{3/4(\Delta y)^2}. \tag{2.3.3}$$

Similarly for the $y$ momentum equation, the term $\partial^2 v/\partial x^2$ needs to be treated in the same fashion.

### 2.4. *The Defect-Correction Procedure*

For calculating convection-dominated flows many workers have opted for hybrid or upwind finite-differences in preference to higher order schemes such as central differences. Two distinct problems lead to this choice. First, use of central differences may result in a "wiggly" solution when $\mathrm{Re}_{\Delta x} > 2$. In practice, when $\mathrm{Re}_{\Delta x}$ varies over the domain, this criterion (based on the maximum local value of $\mathrm{Re}_{\Delta x}$), tends to be too restrictive. Wiggles do not occur unless the maximum local $\mathrm{Re}_{\Delta x}$ is significantly larger (e.g., Kim and Moin [21]). Second, when $\mathrm{Re}_{\Delta x} > 2$, the coefficient matrix of the discretised system is no longer diagonally dominant and classical iteration techniques like Gauss–Seidel may not converge. First-order approximations do not suffer from either of these problems since they effectively limit the maximum $\mathrm{Re}_{\Delta x}$ to two by adding artificial diffusion; however, they accomplish this by sacrificing accuracy. One way around the iterative instability is to modify the iterative method. First-order (upwind) values are used on the left-hand side of the equation and the difference between first- and second-order approximations taken at the previous iteration step is used on the right-hand side. At convergence, the solution is second-order accurate but the convergence properties are similar to those of the first-order approximation. (However, see the results section for a discussion of this point.)

The application of the technique is demonstrated as follows. Assume that the solution of the equation

$$\mathscr{L}(u) = f \tag{2.4.1}$$

is required. Denote by $\mathscr{L}^1_{(h)}(u)$ a first-order approximation and by $\mathscr{L}^2_{(h)}(u)$ a second-order approximation to $\mathscr{L}(u)$ based on a grid of cell size $h$. A second-order accurate solution $\mathscr{L}^2$ can be obtained by the procedure

$$\mathscr{L}^1_{(h)}(u^{n+1}) = f + (\mathscr{L}^1_{(h)}(u^n) - \mathscr{L}^2_{(h)}(u^n)), \tag{2.4.2}$$

where $u^n$ is the solution after the $n$th step. At convergence, $u^{n+1} = u^n = u^\infty$ satisfies

$$\mathscr{L}^2_{(h)}(u^\infty) = f. \tag{2.4.3}$$

Of course, the solution may contain spatial oscillations.

For sufficiently smooth problems, Hackbusch [22] has shown that this proce-

dure yields a second-order accurate solution after the first two defect correction iterations (i.e., after $n=2$). This has been shown to be true by Hemker [17] for solving the Euler equations. However, instead of iterating Eq. (2.4.2) to convergence at each "step," the latter author instead prefers to perform only one multigrid relaxation cycle per defect correction step. This is the approach taken by the present authors.

This "defect-correction" need not be based on central differences. Other approximations such as third-order upwinding, e.g., "QUICK" (Leonard [23]) can be used. In this work only central differences have been tried. Experience shows that the differences in results generated by the latter two methods are small (Avva, private communication [24]).

When this procedure is used in conjunction with the multigrid technique, the correction is directly applied only to the equations on the finest grid. Since the coarse-grid equations are approximations to those on the fine grid, there is no inconsistency. At convergence the solution is second-order accurate.

### 2.5. *Multigrid Implementation*

Recently, considerable material has been written concerning the multigrid technique. Detailed descriptions can be found in various articles including: Brandt [25], Vanka [3–5], Fuchs and Zhao [2], and Hackbusch and Trottenberg [26]. The FAS-FMG (full approximation scheme–full multigrid) algorithm originally developed by Brandt [25] and since used by many investigators, including Vanka [3–5] and Ghia *et al.* [1], is used for this work. This is a generalisation of the original "correction scheme" to non-linear problems. An adaptive or "accomodative" (rather than a fixed) multigrid cycling algorithm is applied. This is also similar to that used by the previously mentioned authors.

The multigrid technique employs a sequence of grids to accelerate the convergence of iterative methods. For this work "standard-coarsening," i.e., doubling the mesh spacing in both the $x$ and $y$ directions from one grid to the next coarsest grid, is used.

The cycle is as follows. Relaxation sweeps are performed on an initial grid (usually the finest). If the smoothing rate is low, i.e.,

$$R^{n+1}/R^n > \mu, \tag{2.5.1}$$

where $R^n$ denotes the average $u$ or $v$ change from the $(n-1)$th step to the $n$th, then the calculation proceeds to the next coarsest grid. (Other workers use different criteria. For instance, Vanka examines the ratio of weighted average residuals from the $x$ and $y$ momentum equations and the continuity equation. Although this is better, no problems were encountered with the above method and the convergence rate is not affected.)

The convergence rate of the multigrid procedure is insensitive to the value chosen for $\mu$. The amount of work required is almost the same for any $\mu$ in the range 0.2–0.8. This can be explained through the following argument.

On each grid level at least two relaxation sweeps must be performed before the smoothing rate can be calculated and relaxation can be continued at a coarser grid level. However, after the first two iterations the smoothing rate usually deteriorates rapidly. Consequently, unless $\mu$ is large (i.e., close to unity), the relaxation process will not continue at the finest grid level for more than two or three relaxation sweeps before switching to the next coarsest level. Therefore, since the number of fine grid relaxation sweeps broadly determines the overall multigrid work requirement, changing $\mu$ over a wide range has little effect. For the calculations reported in this paper, $\mu$ was set at 0.6.

If the convergence rate is slow, so that Eq. (2.5.1) is true, the current approximation to the solution is interpolated, or restricted, to the next coarsest grid. The restriction operator is described in a later section. The residuals of the momentum and continuity equations are also restricted. On the coarse grid ($k$), the equations solved are approximations to the fine grid ($k+1$) equations. They can be symbolically expressed as

$$\mathscr{L}_k U_k = \mathscr{F}_k, \tag{2.5.2}$$

where

$$\mathscr{F}_k = \mathscr{L}_k(I_k^{k+1} U_{k+1}) + I_k^{k+1}(\mathscr{F}_{k+1} - \mathscr{L}_{k+1} U_{k+1}), \tag{2.5.3a}$$

and on the finest grid ($k = M$) the original problem is solved, i.e.,

$$\mathscr{F}_M = f_M. \tag{2.5.3b}$$

Here, $\mathscr{L}_k$ represents the difference operator on grid $k$, $I_k^{k+1}$ is an interpolation operator from grid $k$ to $k+1$, and $U_k$ represents $u$, $v$, or $P$ on grid $k$.

Local relaxation schemes tend to reduce the short wavelength error components much faster than those of longer wavelength. The purpose of the coarse grid iterations is to reduce the long wavelength errors. These can be resolved on the coarser grids on which relaxation sweeps are cheaper. The strategy behind the coarse-grid iteration process is more easily seen if Eq. (2.5.2) is rewritten as

$$\mathscr{L}_k U_k - \mathscr{L}_k(I_k^{k+1} U_{k+1}) = I_k^{k+1}(\mathscr{F}_{k+1} - \mathscr{L}_{k+1} U_{k+1}). \tag{2.5.4}$$

The left-hand side is the difference between the coarse-grid operator acting on the coarse-grid solution and the coarse-grid operator acting on the interpolated fine-grid solution (which is held fixed). Thus Eq. (2.5.2) can be interpreted as an equation for the correction from the interpolated fine-grid solution. The right-hand side is the interpolated residual of the fine-grid equation. After the fine-grid solution has been found the residual will be zero and the solution to Eq. (2.5.4) becomes $U_k = I_k^{k+1} U_{k+1}$ as required. When the residual is non-zero, it acts as a forcing term for the coarse-grid correction. After Eq. (2.5.2) is solved to the necessary accuracy

(this is discussed below), the correction to $U_k$ must be transferred back (prolongated) to the fine grid, i.e.,

$$U_{k+1}^{\text{new}} = U_{k+1}^{\text{old}} + I_{k+1}^{k}(U_k - I_k^{k+1}U_{k+1}^{\text{old}}). \tag{2.5.5}$$

This is vital for the success of the scheme. Changes in the variables are transferred back to finer grids rather than the variables themselves.

The coarse-grid solution is transferred to the fine grid when

$$\varepsilon^k < \delta\varepsilon^{k+1}, \tag{2.5.6}$$

where

$$\varepsilon^k = \frac{1}{\bar{U}}\left(\sqrt{\frac{1}{N_u}\sum (u^n - u^{n-1})^2}\sqrt{\frac{1}{N_v}\sum (v^n - v^{n-1})^2}\right)^{1/2}. \tag{2.5.7}$$

(The superscript $n$ denotes the iteration number and $N_u$ and $N_v$ are the number of $u$ and $v$ gridpoints, respectively. $\bar{U}$ is a velocity scale.) The parameter $\delta$ determines how far a coarse grid solution is converged before being used to correct the solution on the fine grid. Clearly if it is too large, the coarse grid iterations will be ineffective at reducing the long wavelength error components, while if it is too small, some of the coarse grid work will be wasted. The optimum range is quite broad. A value of $\delta = 0.3$ was found to give a good convergence rate over a wide range of Reynolds numbers and grid sizes. Varying $\delta$ over the range (0.1–0.5) has little effect on the overall convergence rate.

Note that on the coarsest grid (which is usually a $2 \times 2$ grid), the same solver is used. Some workers have employed direct methods or Newton iterations; however, this was found unnecessary.

### 2.6. *The Smoothing Operators*

The smoothing operators are very similar to Vanka's [3]. Two types were tried; a block solver and a line-block solver. Brandt [27] suggested that, to obtain a good smoothing rate with the multigrid approach, it is necessary for the unknowns be updated in a locally coupled manner. Brandt's DGS technique updates the continuity equation separately, but applies corrections that ensure continued satisfaction of the $x$ and $y$ momentum equations at each point. Vanka's procedure is slightly different. It solves for four velocity components (at the four cell faces) and the pressure at the centre of the cell simultaneously for each cell. This scheme has very good smoothing properties.

The first scheme is a variant of Vanka's SCGS scheme. The $x$ and $y$ momentum equations for the four cell faces (cf. Fig. 1) together with the continuity equation about the cell centre can be expressed as:

$$\sum_{|k|+|l|\leqslant 1} A_{i+k,j+l}^{u_{i,j}} u_{i+k,j+l} + \Delta y(P_{i,j} - P_{i-1,j}) = 0, \tag{2.6.1a}$$

$$\sum_{|k|+|l|\leq 1} A^{u_{i+1,j}}_{i+k,j+l} u_{i+1+k,j+l} + \Delta y(P_{i+1,j} - P_{i,j}) = 0, \tag{2.6.1b}$$

$$\sum_{|k|+|l|\leq 1} A^{v_{i,j}}_{i+k,j+l} v_{i+k,j+l} + \Delta x(P_{i,j} - P_{i,j-1}) = 0, \tag{2.6.1c}$$

$$\sum_{|k|+|l|\leq 1} A^{v_{i,j+1}}_{i+k,j+l} v_{i+k,j+1+l} + \Delta x(P_{i,j+1} - P_{i,j}) = 0, \tag{2.6.1d}$$

$$(u_{i+1,j} - u_{i,j})\,\Delta y + (v_{i,j+1} - v_{i,j})\,\Delta x = 0. \tag{2.6.1e}$$

The $A$'s are functions of the velocity components: the current values are used. The $A$'s are given by Eq. (2.2.2a)–(2.2.2e), where the following associations have been made: $A_{i,j} = A_c$, $A_{i,j+1} = A_n$, $A_{i+1,j} = A_e$, $A_{i,j-1} = A_s$, and $A_{i-1,j} = A_w$. These equations can be written:

$$\begin{bmatrix} A^{u_{i,j}}_{i,j} & A^{u_{i,j}}_{i+1,j} & 0 & 0 & \Delta y \\ A^{u_{i+1,j}}_{i-1,j} & A^{u_{i+1,j}}_{i,j} & 0 & 0 & -\Delta y \\ 0 & 0 & A^{v_{i,j}}_{i,j} & A^{v_{i,j}}_{i,j+1} & \Delta x \\ 0 & 0 & A^{v_{i,j+1}}_{i,j-1} & A^{v_{i,j+1}}_{i,j+1} & -\Delta x \\ -\Delta y & \Delta y & -\Delta x & \Delta x & 0 \end{bmatrix} \cdot \begin{bmatrix} u'_{i,j} \\ u'_{i+1,j} \\ v'_{i,j} \\ v'_{i,j+1} \\ P'_{i,j} \end{bmatrix} = \begin{bmatrix} R^{u_{i,j}} \\ R^{u_{i+1,j}} \\ R^{v_{i,j}} \\ R^{v_{i,j+1}} \\ 0 \end{bmatrix}, \tag{2.6.2}$$

where the $R^{u,v}$'s represent the remaining terms from Eq. (2.6.1a)–(2.6.1e). (This equation differs from Vanka's only by the inclusion of the off-diagonal terms in the first four rows. It was expected that this would speed up convergence slightly since the systems is "more implicit." Tests indicate that the actual convergence rate is similar for the two variations.)

The system (2.6.2) can be solved easily using Gaussian elimination. The velocity components and the pressure are updated using under-relaxation:

$$(u, v)^{\text{new}} = (u, v)^{\text{old}} + r_{(u,v)}((u, v)' - (u, v)^{\text{old}}) \tag{2.6.3a}$$

$$P^{\text{new}} = P^{\text{old}} + r_P(P' - P^{\text{old}}). \tag{2.6.3b}$$

The relaxation coefficient $r_{(u,v)}$ may depend on Reynolds number and on whether the defect correction is applied or not. The optimum value for $r_P$ is always very close to unity. All results presented in this paper were obtained with $r_P = 1$. This method is called symmetrically coupled Gauss–Seidel (SCGS).

Approximately optimal values for $r_{(u,v)}$ are given in Table III (see results section). The optimum value is a function of the Reynolds number but only weakly dependent on the grid size. Figure 2 shows the time required to reach convergence as a function of $r_{(u,v)}$ for driven cavity flow at $\text{Re} = 1000$ for a $16 \times 16$ grid.

In the second scheme the variables for a whole line of cells are updated simultaneously. If the sweep is in the $y$ direction and the variables are ordered in the following way, $u_{i-1,j}$, $v_{i,j}$, $P_{ij}$, $v_{i,j-1}$, $u_{i,j}$, $v_{i+1,j}$, $P_{i+1,j}$, $v_{i+1,j-1}$, $u_{i+2,j}$, $v_{i+2,j}$, $P_{i+2,j}$, $v_{i+2,j-1}$, $u_{i+3,j}$, ..., the resulting matrix system has a banded structure. This system was solved by a specially written banded-matrix solver designed

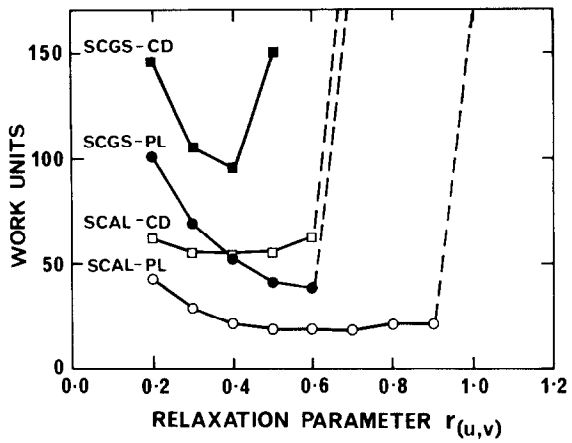


FIG. 2. Work units required for convergence as a function of relaxation parameter ($r_{(u,v)}$). A $16 \times 16$ grid was used at $Re = 1000$.

to take advantage of the sparsity of the bands. This line-block solver was used with alternating zebra-sweeping. The order of sweep was: even horizontal lines, odd horizontal lines, even vertical lines, odd vertical lines. This approach, called symmetrically coupled alternating line (SCAL), was expected to be more robust when the cell aspect ratio is far from unity and the flow is predominantly uni-directional. Indeed, this technique is more robust than SCGS for the backward-facing step flow; for driven cavity flow the difference is not significant. It is important to note that the cost of using the line solver is only about 50% larger than for the SCGS solver per sweep (cf. Table I). With the line solver the multigrid method is less sensitive to the relaxation coefficient ($r_{(u,v)}$) (cf. Fig. 2). Both methods were applied to the power-law (PL) and central-difference (CD) defect-correction discretizations.

### 2.7. *Prolongation and Restriction*

The routines for passing information between grids were written so that the effects of altering the order of the interpolation could be studied. Two-dimensional

TABLE I

CPU-times per Sweep for a $32 \times 32$ Grid on a VAX-11-750 Computer

| Solution method | CPU-time (s) |
|---|---|
| SCGS-PL | 5.3 |
| SCGS-CD | 5.7 |
| SCAL-PL | 7.3 |
| SCAL-CD | 9.5 |

*Note.* The scheme was coded in double precision.

Lagrange interpolation with four, nine, and 16 points, giving errors $O(\Delta x^2, \Delta y^2)$, $O(\Delta x^3, \Delta y^3)$, and $O(\Delta x^4, \Delta y^4)$ were used. At the edges of the domain, the distribution of the interpolation points is not symmetrical for the higher order schemes. As seen from Table II, the use of more accurate higher order interpolation schemes actually degrades the convergence rate. The table gives the equivalent number of fine-grid iterations for the SCGS-PL and SCGS-CD schemes for different restriction and prolongation operators. Only tests using second- and third-order interpolation are listed: the fourth-order approach has even worse convergence characteristics.

The tests indicate that it is especially bad to use high-order interpolation for the coarse to fine grid transfer (prolongation). This may be due to the introduction of high-frequency noise that the fine grid iterations have to remove. There is no significant difference between the results for the 4- and 9-point restriction operators. All further results were obtained using 4-point Lagrange interpolation (equivalent to bilinear interpolation) for both prolongation and restriction. In the future, weighted-average restriction operators (which are the adjoints of the prolongation operators) will be assessed (e.g., see [1, 7]).

### 2.8. *The Adaptive Grid Technique*

The adaptive technique used is based on the work of Caruso [13] who in turn, modified the approach of Berger [28]. It uses a sequence of overlapping grids of increasing fineness. The adaption criterion is based on the truncation error; the solution error is also monitored. The grid can be refined until the solution error estimate is below some desired level. The procedure is described below.

The solution is computed on an intermediately fine grid spanning the whole domain and stored for later use. Following this, the problem is solved on a grid

TABLE II

The Effect of Interpolation Order on the Convergence Rate of the Multigrid Method

| Prolongation/ Restriction order | Scheme | $r_{(u,v)}$ 0.2 | 0.3 | 0.4 | 0.5 | 0.6 |
|---|---|---|---|---|---|---|
| (2, 2) | SCGS-PL | 100.70 | 68.73 | 51.91 | 41.00 | 38.17 |
| (3, 3) | SCGS-PL | | > 100 | > 150 | | |
| (3, 2) | SCGS-PL | | 75.63 | 54.28 | 42.56 | > 100 |
| (2, 2) | SCGS-CD | 145.86 | 105.44 | 95.28 | Div | |
| (2, 3) | SCGS-CD | | | 160.17 | Div | |
| (3, 3) | SCGS-CD | | | > 200 | Div | |
| (3, 2) | SCGS-CD | | 142.50 | 118.22 | Div | |

*Note.* The values indicate the work required to achieve convergence. A $16 \times 16$ grid was used with $\mathrm{Re} = 1000$. The convergence criterion was $\max(\Delta u, \Delta v) < 10^{-4}$.

with twice the grid spacing in both the $x$ and $y$ directions. From these two solutions, the solution error can be estimated by the Richardson method.

It is assumed that the solution error can be expressed as a Taylor series

$$\begin{aligned} e(\mathbf{x}; h) &= u(\mathbf{x}; 0) - u(\mathbf{x}; h) \\ &= h^p F(\mathbf{x}) + h^q G(\mathbf{x}) + \cdots, \end{aligned} \tag{2.8.1}$$

where $u(\mathbf{x}; 0)$ is the exact solution, $h$ is the mesh size, and $p$ is the order of the method. This expansion is valid for solutions with continuous derivatives. For elliptic systems, this is the case.

If the grid spacing is doubled in both $x$ and $y$, the solution error becomes

$$\begin{aligned} e(\mathbf{x}; 2h) &= u(\mathbf{x}: 0) - u(\mathbf{x}; 2h) \\ &= (2h)^p F(\mathbf{x}) + (2h)^q G(\mathbf{x}) + \cdots, \end{aligned} \tag{2.8.2}$$

Subtracting Eq. (2.8.1) from (2.8.2) and dividing the result by $2^p - 1$ gives an estimate of the solution error

$$\begin{aligned} \tilde{e}(\mathbf{x}; h) &= \frac{u(\mathbf{x}; h) - u(\mathbf{x}; 2h)}{2^p - 1} \\ &= h^p F(\mathbf{x}) + h^q \left( \frac{2^q - 1}{2^p - 1} \right) G(\mathbf{x}) + \cdots, \end{aligned} \tag{2.8.3}$$

Then comparing Eqs. (2.8.1) and (2.8.3) gives

$$\tilde{e}(\mathbf{x}; h) = e(\mathbf{x}; h) + O(h^q), \tag{2.8.4}$$

so the estimate is accurate to order $q$.

This approach can be applied to estimate the solution error for each of the three variables $u$, $v$, and $P$. However, as a staggered grid is used, the locations of the variables on the coarse grid do not coincide with those on the fine grid so interpolation to fine-grid locations is required. Also, the primitive equations do not fix the pressure absolutely but only determine it to within an arbitrary constant; only pressure differences are important physically. Therefore solution error estimates for the pressure cannot be accurate.

The truncation error is defined by

$$\tau(\mathbf{x}; h) = \mathscr{L}_h[u(\mathbf{x}; 0), v(\mathbf{x}; 0), (P(\mathbf{x}; 0))] - f, \tag{2.8.5}$$

where $\mathscr{L}_h[\cdots] = f$ represents the $u$ or $v$ momentum equation or the continuity equation. However, as the exact solution is unknown, the true solutions are first estimated by

$$\tilde{u}(\mathbf{x}; 0) = u(\mathbf{x}; h) + \tilde{e}(\mathbf{x}; h), \tag{2.8.6}$$

and used in place of $u(\mathbf{x}; 0)$..., to give

$$\tilde{\tau}(\mathbf{x}; h) = \mathscr{L}_h[\tilde{u}(\mathbf{x}; 0), \tilde{v}(\mathbf{x}; 0), (\tilde{P}(\mathbf{x}; 0))] - f. \tag{2.8.7}$$

In this way the truncation errors for the three governing equations are estimated.

For each equation, the grid points at which the truncation error estimate is above some predetermined value (say $\tau^*$) are marked. The marked gridpoints are then clustered into groups. A point is added to a group if it is within certain $x$ and $y$ distances ($d_x$ and $d_y$) of any other point within the group. (This operation can be expensive unless the number of marked gridpoints is small.) Once the clustering is completed, boundary aligned rectangles enclosing all grid points of each group are drawn. The boundaries are extended by a safety margin from the $x$ and $y$ extremes of the enclosed points. A check to ensure that these rectangles do not overlap is made. If overlapping occurs, the offending rectangles are combined into a new rectangle. Checking for overlapping continues and rectangles are combined, until the set of rectangles is disjoint. These rectangles are then covered with grids of spacing half that of the previously finest grid. (Only boundary aligned subgrids are considered.) A method in which the regions are chosen in accord with an estimate of their effect on the solution will be tried in the future.

The solution is then computed on the combination of the new subgrids and the grid at the previous level. At convergence, the solution on the overlapped component of the coarse grid should coincide with the fine grid solution. This can be achieved by adding forcing terms to equations on the coarse grid in the overlap region(s).

Suppose that the equations to be solved on the coarse grid can be represented by

$$\mathscr{L}_{2h}[u(\mathbf{x}; 2h), v(; 2h), (P(\mathbf{x}; 2h))] = f. \tag{2.8.8a}$$

When overlapping occurs these are modified to

$$\mathscr{L}_{2h}[u(\mathbf{x}; 2h), \ldots] = \mathscr{L}_{2h}[u'(\mathbf{x}; h), v'(\mathbf{x}; h), (P'(\mathbf{x}; h))], \tag{2.8.8b}$$

where $u'(\mathbf{x}; h)$, $v'(\mathbf{x}; h)$, and $P'(\mathbf{x}; h)$ are obtained by interpolating the fine-grid variables to the coarse-grid variable locations. This correction is very similar to the multigrid correction and provides the basis of the adaptive-multigrid approach. (For linear equations it is possible to demonstrate exact equivalence at convergence.) The fine-grid boundary values are obtained from the coarse grid. However, the solution at these points changes as the iteration proceeds because, although the truncation error criterion is satisfied there, the solution error may be large. Thus Eqs. (2.8.8) are used to update the solution variables on the coarse grid and iteration between the coarse and fine grids is performed.

The choice of the clustering distances $d_x$ and $d_y$, together with $\tau^*$, determines the part of the domain covered by the refined grid components. Choosing $d_x$ and $d_y$ to be very small can result in the mesh structure at the new level becoming fragmented, especially if the truncation error estimate is noisy. Very large values

of $d_x$ and $d_y$ result in a single refined grid component covering the entire domain. For the results described in Section 3, $d_x$ and $d_y$ vary with level. They are set to twice the $x$ and $y$ cell widths of the currently finest grid, respectively.

Overall convergence requires the use of "mass-conservative" interpolation to evaluate the $u$ and $v$ boundary values for the subgrids. The mass fluxes into, and out of, subgrids must be consistent with values on the "parent" grids. However, after any iteration sweep continuity is not necessarily satisfied on a cell by cell basis; this is only true at convergence.

To clarify this, consider Fig. 3 which shows a section of the grid at an internal boundary. If linear interpolation is used, the mass flux will be consistently transferred during the fine to coarse interpolation. The mass flux into coarse-grid cell $i$, which is proportional to $v_i^c \Delta x^c$, should equal the flux into the fine-grid cells $j$ and $j+1$, which is proportional to $(v_j^f + v_{j+1}^f)\Delta x^f$. For a uniform grid, this reduces to

$$v_i^c = \tfrac{1}{2}(v_j^f + v_{j+1}^f), \tag{2.8.9}$$

which is just linear interpolation. The coarse to fine grid transfer, however, does not guarantee conservation. In that case linear interpolation gives, for example,

$$v_j^f = \tfrac{3}{4}v_i^c + \tfrac{1}{4}v_{i-1}^c. \tag{2.8.10}$$

Neither global nor cell by cell conservation is satisfied. Various modifications are used to ensure at least global conservation. The method adopted here is to use linear (or higher order) interpolation as a first approximation and then to correct the velocities so that conservation is satisfied on a cell by cell basis. This is done by insisting that Eq. (2.8.9) holds for the coarse to fine transfer as well, i.e.,

$$\tfrac{1}{2}(v_j^f + v_{j+1}^f) = v_i^c. \tag{2.8.11}$$

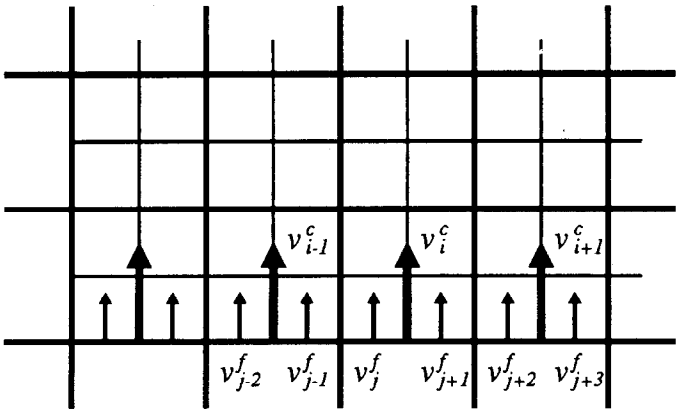


FIG. 3. Interpolation at a coarse-grid/fine-grid interface. For details see the text.

Thus the interpolated fine-grid vertical velocity components can be calculated from

$$v_j^f = \left[ \frac{2v_i^c}{v_j^{f\prime} + v_{j+1}^{f\prime}} \right] v_j^{f\prime} \tag{2.8.12a}$$

and

$$v_{j+1}^f = \left[ \frac{2v_i^c}{v_j^{f\prime} + v_{j+1}^{f\prime}} \right] v_{j+1}^{f\prime}, \tag{2.8.12b}$$

where $v_j^{f\prime}$ and $v_{j+1}^{f\prime}$ are intermediate estimates. The same can be done at the vertical boundary for the horizontal velocity components.

This procedure may seem somewhat ad hoc (or least lacking a strong theoretical foundation) and one has to wonder about its effect on the accuracy of the adaptive-multigrid results. Two points need to be borne in mine. First, if the initial interpolation is done with a higher order scheme, it is expected (and indeed found) that mass conservation between grids nearly is satisfied, and the adjustment is very small. Second, at the boundaries of the refined grids, the coarse grid gives adequate resolution anyway; this is part of the refinement criterion. Thus the interpolation procedure does not degrade the accuracy significantly.

To make the computed code efficient with minimal complication, attention must be paid to the data structures. The framework used was developed by Berger [28] and modified for elliptic problems by Caruso [13]. Details can be found in an article by Berger [29].

### 2.9. *Combined Adaptive-Multigrid Method*

The adaptive grid method just described fits in very well with the multigrid solution process. The addition of finer mesh components spanning only parts of the domain was described early in multigrid history (e.g., Brandt [25], who called it segmental refinement), but the idea has not been put into practice often. Bai [30] applied segmental refinement to Poisson's equation in two dimensions. However, he specifies the local refinements a priori so his method is not truly adaptive. He suggests that automatic refinement can be done straightforwardly. Becker and Trottenberg [31] considered 2-dimensional subsonic potential flow around an aerofoil. They used segmental refinement with the refinement criterion depending on the local discretization error. Very good smoothing rates were achieved.

To describe the adaptive-multigrid algorithm it is probably easiest to use an example. Consider driven cavity flow (cf. Section 3). Before starting, a grid level $S$ is chosen after which adaptive refinement will be allowed. The $S$ grid should be sufficient so that the solution and truncation errors, based on the solution at this and the previous grid level, are reasonably smooth. (This step is not a necessary part of the adaptive approach. The main reason for it was to ensure that there would be a solution covering the entire domain which could be used for contouring.) The algorithm is then as follows.

First, the solution is found on a very coarse grid (e.g., a $2 \times 2$ grid). This solution is stored and interpolated to the next finest grid (a $4 \times 4$ grid, for this example). The multigrid method is used to solve the problem on the combined grid system (the $2 \times 2$ and the $4 \times 4$ grids). This solution is also stored. The procedure continues until the solution is found at level $S$. Then the truncation error estimates for the three equations are evaluated and used in the refinement process. Clustering of grid points which have truncation errors above the threshold is carried out and fine grid patches are added to the grid system. Multigrid is then used to find the solution on this composite grid system. If there is more than one component grid at the last level, a smoothing sweep consists of one sweep over each subgrid. The smoothing rate (defined by Eq. (2.5.1)), which determines whether to move to a coarser grid level, is based on the average $u$ and $v$ changes on the set of subgrids. The procedure can be considered a multigrid process in which the fine grid covers only part of the domain; the solution in the remainder of the domain can be obtained by interpolating the coarse-grid solution. (However, note that the "coarse-grid solution" changes as finer levels are added.) It is the limited extent of the fine grid that provides the efficiency of the method.

If the injection and prolongation are done consistently and care is taken to make sure that mass is conserved at grid boundaries, the multigrid process is convergent. After convergence on a given set of grids, a new estimate is made of the solution and truncation errors using the current solution and the previously stored solution. Further grid refinement is allowed only on the previously refined grids. Thus, new subgrids lie entirely within previous subgrids. This procedure continues automatically until the solution error is reduced to a satisfactory level (or until the available computer memory is exhausted). This method can be regarded as a standard multigrid method in which the solution on part of the grid is obtained by interpolation from the coarser grid. Since the greatest part of the cost of multigrid methods is associated with fine grid iterations and the finest grid occupies only a small portion of the solution domain, this method has considerable advantages in both speed and memory.

An important additional refinement needs to be made when using the central-difference defect correction. When using the standard multigrid method, the grid must be sufficiently fine to achieve adequate resolution of the solution, otherwise the convergence rate is impaired considerably. This is especially true for high Re cases (see the next section.) With the adaptive scheme described above, this is a problem since intermediate solutions are computed on coarse grids. This shortcoming can be circumvented by not iterating to convergece on the intermedciate grids. Only a fixed amount of work is performed at each grid-level and, if convergence is not achieved, then the solution process continues on the next grid level. Convergence is required only at the highest grid levels. Naturally, this modification makes the intermediate error estimates unreliable. However, in practice it is not a problem. The convergence rate is usually rapid initially but slows as the solution nears convergence. The most important consideration is that refined grids are inserted where they are needed. Providing the inaccurate error estimates results in these refinement regions

being larger than they would be otherwise, which is the likely outcome, there will be no detrimental effect to the accuracy of the final solution.

## 3. Results

Results have been obtained for two flows; driven cavity and backward-facing step flow. Both these cases have been studied extensively and are well documented in the literature [1, 3, 4, 13, 21, 32, 33–36]. The main emphasis of the results will be with regard to the convergence properties of the method rather than the details of the solutions.

Some results obtained with a "pure" multigrid method will be examined first. This will establish a benchmark which can be used to examine the improvements in efficiency resulting from adaptive refinement. Results from an adaptive-grid method (effectively equivalent to embedded refinement without multigrid) can be found in Caruso [13].

### 3.1. *Driven Cavity Flow: Multigrid Results*

Driven cavity flow has long been a standard test problem for Navier-Stokes codes. In the last ten years accurate results have been obtained by various authors [1, 3, 6, 32, 33, 34] and there is general agreement on the solutions for Reynolds numbers up to about 10,000 in two dimensions. Recently, experimental results have become available (e.g., [35, 37]). They indicate that for moderately low Reynolds numbers ($\mathrm{Re} \sim 1000$) the flowfield becomes 3-dimensional with the appearance of Taylor–Görtler vortices. Some 3-dimensional numerical simulations have predicted these structures [4, 21, 35]. Consequently, high Reynolds number 2-dimensional numerical simulations are now only of interest as a test problem for Navier–Stokes codes.

The numerical setup consists of a 2-dimensional square cavity with zero velocity boundary conditions except at the top where $u$ is a constant. The streamline pattern for the 2-dimensional cavity at $\mathrm{Re} = 5000$ is shown in Fig. 4. The top of the cavity moves at a constant speed ($u = 1$) to the right. This figure shows the primary eddy, the three secondary eddies, and the beginnings of tertiary eddies at the two lower corners. Details on eddy positions, reattachments lengths, etc. can be found in Refs. [1, 32, 33].

Table III presents some results obtained with the non-adaptive multigrid method for driven cavity flow for three different Reynolds numbers ($\mathrm{Re} = 100$, 1000, and 5000) and for grid sizes from $8 \times 8$ to $256 \times 256$. The entries are the number of *work units* to reach convergence. (A work unit is the equivalent amount of computational work needed to do one iteration sweep on the finest grid using the multigrid smoother and is the standard way of presenting multigrid convergence results.) The convergence criterion required the maximum change in $u$ or $v$ between fine-grid iteration sweeps to be less than $10^{-4}$. Typically, the average change is around $10^{-5}$. The table also indicates the value of the relaxation parameter $r_{(u,v)}$. Several impor-

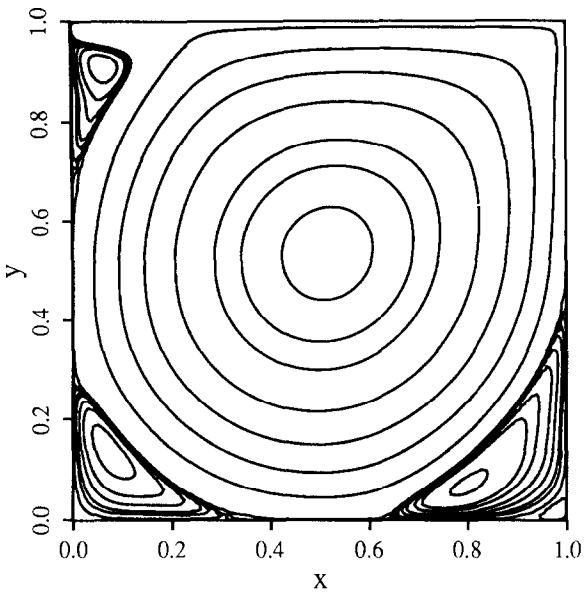


FIG. 4. Streamlines for Re = 5000 cavity flow.

TABLE III

Work Units Required to Obtain Convergence for the Multigrid Scheme.

| Re | Scheme ($r_{(u,v)}$) | $8 \times 8$ | $16 \times 16$ | $32 \times 32$ | $64 \times 64$ | $128 \times 128$ | $256 \times 256$ |
|---|---|---|---|---|---|---|---|
| 100 | SCGS-PL (0.6) | 17.75 | 18.72 | 19.88 | 20.35 | 19.24 | |
| | SCAL-PL (0.6) | 13.00 | 15.44 | 22.16 | | 28.89 | |
| | SCGS-CD (0.5) | 21.56 | 24.53 | 21.66 | 19.67 | 19.40 | |
| | SCAL-CD (0.5) | 15.50 | 16.56 | 28.70 | | 27.67 | |
| 1000 | SCGS-PL (0.6) | 23.44 | 38.17 | 48.78 | 43.18 | 42.77 | |
| | SCAL-PL (0.6) | 14.25 | 18.88 | 23.19 | | 29.24 | 27.93 |
| | SCGS-CD (0.4) | 72.19 | 95.28 | 114.05 | 66.58 | 44.30 | |
| | SCAL-CD (0.4) | 41.25 | 54.31 | 88.11 | | 46.99 | |
| 5000 | SCGS-PL (0.3) | | 88.13 | 156.71 | | | |
| | SCAL-PL (0.5) | | 23.50 | | 41.50 | 50.65 | |
| | SCGS-CD (0.2) | | 324.14 | | | | |
| | SCAL-CD (0.2) | | 111.09 | | | 140.56 | 72.39 |

*Note.* Work units are equivalent to fine grid iterations. The convergence criterion was $\max(\Delta u, \Delta v) \leqslant 10^{-4}$.

tant trends are evident. Typically, for single grid solvers at the higher Reynolds numbers, hundreds or even thousands of fine grid iterations are required to obtain a solution. On the finest grids for each Reynolds number, the work required to obtain a solution by using multigrid increases from about 20 units for $Re = 100$ to about 70 for $Re = 5000$. This shows both the efficiency of the multigrid approach and its relative insensitivity to the Reynolds number. The table also indicates that the number of work units required for convergence is only a weak function of the grid size, a result consistent with the theoretical prediction that the number of work units should not depend upon the size of the grid. For the higher Re cases, with the central-difference smoothers and large enough grid sizes, the number of work units actually decreases significantly as the grid size is increased. This effect was mentioned by Ghia *et al.* [1]. It can be seen for the $Re = 1000$ case using the SCGS-CD scheme for grid sizes $32 \times 32$, $64 \times 64$, and $128 \times 128$. A possible explanation is that the grid spacing must be sufficiently fine to resolve the fine-scale structure. Otherwise, the solution may be slightly "wiggly" and convergence may be hindered. (However, central difference wiggles could only be detected for the coarsest grids for each Reynolds number.) Note that although the central-difference solutions are more expensive to obtain for a given grid size, they are considerably cheaper when compared in terms of equal accuracy.

Table IV gives the value of the streamfunction at the centre of the primary eddy ($\Psi^P_{\min}$) for the upwind and centred difference schemes at the same Reynolds numbers and grid sizes used in the previous table. This parameter is a fairly sensitive indicator of the accuracy of the solution. These values can be compared to those obtained by Ghia *et al.* [1] who used a second-order vorticity-streamfunction approach with a grid size of $128 \times 128$ for the $Re = 100$ and 1000 cases, and $256 \times 256$ for the $Re = 5000$ case. (For the finest grids used for each Reynolds number, other parameters such as secondary and tertiary vortex strengths and reat-

TABLE IV

Values of the Streamfunction at the Centre of the Primary Vortex

| Re | Scheme | $8 \times 8$ | $16 \times 16$ | $32 \times 32$ | $64 \times 64$ | $128 \times 128$ | $256 \times 256$ | $\Psi^{P_*}_{\min}$ |
|---|---|---|---|---|---|---|---|---|
| 100 | PL | −0.0843 | −0.0956 | −0.1010 | −0.1028 | −0.1034 | | −0.1034 |
| | CD | −0.0945 | −0.1002 | −0.1025 | −0.1033 | −0.1034 | | −0.1034 |
| 1000 | PL | −0.0440 | −0.0619 | −0.0807 | −0.0995 | −0.1114 | −0.1167 | −0.1179 |
| | CD | −0.0718 | −0.0908 | −0.1054 | −0.1150 | −0.1178 | | −0.1179 |
| 5000 | PL | −0.0229 | −0.0349 | −0.0487 | −0.0657 | −0.0843 | | −0.1190 |
| | CD | | −0.0639 | | | −0.1139 | −0.1181 | −0.1190 |

*Note.* This table gives a comparison of the first (PL) and second (CD) methods as a function of grid size. $\Psi^{P_*}_{\min}$ are values taken from Ghia *et al.* [1] who used a second-order streamfunction vorticity approach.

tachment positions agree very well with the values given in [1].) This table shows that the central difference solution converges much more rapidly (with grid size) than the upwind solution. For the higher Reynolds number cases, it is necessary to use approximately 16 times the number of grid points with an upwind scheme to achieve the same accuracy obtained with a central difference scheme.

### 3.2. *Driven Cavity Flow: Multigrid with Adaptive Refinement*

The advantage of using adaptive gridding is twofold. First, there is a reduction in the amount of work required to achieve convergence and, second, the amount of computer memory required to obtain a solution to the same accuracy is considerably smaller. These factors are examined in Table V.

This table compares adaptive-multigrid results with pure multigrid results. The runs were done with the SCGS-CD smoother for $Re = 100$ and 1000, and the SCAL-CD smoother for $Re = 5000$. The effect of using different values of the truncation error refinement criterion ($\tau^*$) is shown. For $Re = 100$ and 1000, no segmental refinement was allowed until after the $32 \times 32$ grid, and for $Re = 5000$, the $64 \times 64$ grid. In all cases listed only two adaptive refinement levels were used.

TABLE V

Convergence Results for the Adaptive-Multigrid Code

| | | $\tau^* = 0.10$ | $\tau^* = 0.05$ | $\tau^* = 0.01$ | Uniform grid multigrid result |
|---|---|---|---|---|---|
| $Re = 100$ | Strength of primary vortex | −0.1030 | −0.1031 | −0.1033 | −0.1034 |
| | Equivalent work units | 4.8 | 5.7 | 13.5 | 19.4 |
| | % memory required | 15 | | 55 | 100 |
| $Re = 1000$ | Strength of primary vortex | −0.1173 | −0.1176 | | −0.1178 |
| | Equivalent work units | 17.0 | 23.5 | | 44.3 |
| | % memory required | 20 | 43 | | 100 |
| $Re = 5000$ | Strength of primary vortex | −0.1179 | | | −0.1181 |
| | Equivalent work units | 51.5 | | | 72.4 |
| | % memory required | 26 | | | 100 |

*Note.* No adaptive refinement allowed until after the $32 \times 32$ grid for $Re = 100$, 1000, and the $64 \times 64$ grid for the $Re = 5000$ cases. In each case two refinement levels were used.

This was done to limit computational costs and so that a direct comparison could be made with computed uniform fine-grid solutions. The uniform multigrid results are taken from Tables III and IV. The parameter used to indicate the accuracy of the adaptive multigrid results is again the primary vortex strength $\Psi^P_{\min}$.

Figure 5 shows the grids used for the $\mathrm{Re} = 5000$ calculation. From the "base" $64 \times 64$ grid level refinement is necessary at the two top corners (where there are singularities) and in the regions occupied by the two secondary eddies at the lower corners. The grid refinement was done automatically.

Consider the results for $\mathrm{Re} = 1000$ obtained with a refinement criterion of $\tau^* = 0.10$. The value obtained for $\Psi^P_{\min}$ for this run was $-0.1173$, while pure multigrid calculations gave $-0.1054$, $-0.1150$, and $-0.1178$ respectively for grid sizes $32 \times 32$, $64 \times 64$, and $128 \times 128$. (Recall that adaptive refinement was allowed only for two levels after the $32 \times 32$ grid level.) This comparison indicates that the adaptive multigrid result is almost as accurate as the result obtained with pure multigrid using a $128 \times 128$ grid. However, the amount of memory and the CPU-time required are reduced to about 20% and 40%, respectively. The advantage becomes greater as higher accuracy is demanded or the Reynolds number is increased. Three-dimensional calculations would also have greater gains.

### 3.3. *Backward-Facing Step Flow: Multigrid with Adaptive Refinement*

The flow over a backward-facing step is another heavily studied numerical test case (e.g., [36, 21]). This is a 2-dimensional flow in a channel which has a sudden expansion on one side. The velocity profile upstream of the step is parabolic and the step height is half the channel height. Figure 6 shows the backward-facing step

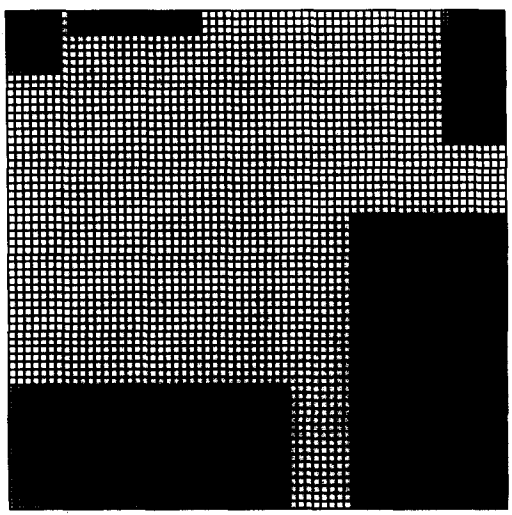

FIG. 5. Composite adaptive grid used for the $\mathrm{Re} = 5000$ calculation. The base grid was $64 \times 64$ and two refinement levels were allowed.

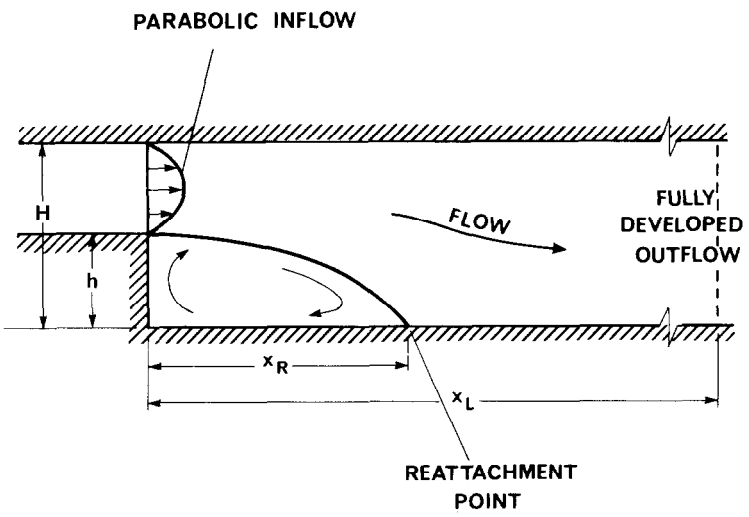


FIG. 6. Backward facing step flow geometry and definition of the parameters involved.

geometry and defines the flow parameters. Experimental results for this particular flow configuration have been given by Armaly *et al.* [38] and numerical results can be found in a number of papers [13, 21, 36]. There is a good agreement between the computation and experiment at low Reynolds numbers ($\mathrm{Re} = \mathbf{u}H/\nu \leqslant 400$, where $\mathbf{u}$ is the mean inflow velocity, $H$ is the channel height, and $\nu$ is the viscosity) but for higher Re, the results begin to deviate. Apparently this is due to 3-dimensional effects [21].

Table VI shows the calculated reattachment lengths (normalized by the step

TABLE VI

Reattachment Length $x_R$ as a Function of Reynolds Number for the Backward Facing Step Problem

| | | | $x_R$ at level | | | $x_R^*$ | | |
|---|---|---|---|---|---|---|---|---|
| Re | $\tau^*$ | $x_L$ | 5 | 6 | 7 | | $\bar{\varepsilon}_u$ | Work units |
| 133 | 0.05 | 12.0 | 4.0 | 4.0 | | 3.9 | 0.002 | 10.2 |
| 267 | 0.05 | 20.0 | 6.5 | 6.5 | | 6.5 | 0.002 | 11.3 |
| 400 | 0.01 | 27.0 | | 8.5 | 8.7 | 8.8 | 0.005 | 10.0 |
| 600 | 0.01 | 36.0 | | 10.1 | 10.7 | 10.8 | 0.02 | 37.0 |
| 800 | 0.01 | 40.0 | | 10.4 | 11.9 | 12.1 | 0.04 | 130.0 |

*Note.* $x_R^*$ are values obtained by Caruso [13] using a second order adaptive grid approach. The work required to obtain these results is given. $\bar{\varepsilon}_u$ is the estimated average solution error for $u$.

height) for Reynolds numbers up to 800. The coarsest grid was $8 \times 2$ in all cases. Hence the grid size at grid level 5, for example, would be $128 \times 32$. Adaptive refinement was only allowed at the final two levels for these runs. The parameter $x_L$ gives the extent of the grid in the $x$ direction. These results were obtained with the SCAL-PL smoother. (Unfortunately some convergence difficulties were experienced with the central-difference smoothers when the Reynolds number exceeded 400. This is currently under investigation.) It can be seen that the predicted reattachment lengths ($x_R$) are very close to the values obtained by Caruso [13] (given by $x_R^*$ in the table). The latter were obtained with a central-difference scheme using adaptive gridding without multigrid and agree with results of Kim and Moin [21] who used a central-difference scheme on a $101 \times 101$ grid.

Table VI also gives the estimated average error in $u$ obtained from the calculations. Even though the grids are very fine, for the higher Reynolds number runs, the average $u$ errors are still fairly large. Furthermore the reattachment length is still changing significantly at the final refinement levels. For instance, for $\mathrm{Re} = 800$, the estimated value of $x_R$ changes from 10.4 at level 6 (a $256 \times 64$ grid), to 11.9 at level 7 (equivalent to a $512 \times 128$ grid). It would be expected that the values of $u$ close to reattachment change significantly between these two solutions, which is indeed the case. Since the finest grid solutions are close to the true solutions (as can be seen from a comparison of the $x_R$'s with literature values), these solution error estimates are probably overestimates.

Figure 7 shows the streamline plots for Reynolds numbers 133, 267, 400, and 600. At $\mathrm{Re} = 600$ a secondary separation bubble is clearly visible on the top wall of the

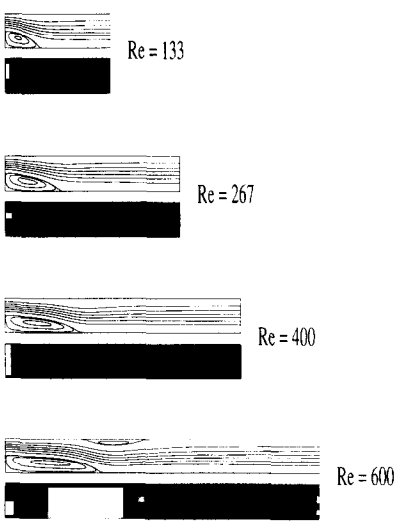


FIG. 7. Streamlines for the backward-facing step problem for a range of Reynolds numbers. The grid systems used in obtaining the solutions are also shown. (The areas covered by the final three levels are shown in black, grey, and white, respectively.)

duct. This has also been found by Kim and Moin [21] and Caruso [13]. The computed solutions also show a very weak bubble for the $\mathrm{Re} = 400$ case. At $\mathrm{Re} = 800$ the secondary separation bubble is longer and stronger.

Figure 7 also shows the structure of the grid systems used in obtaining these solutions. The final three levels are shown as black, grey, and white, respectively. For all four Reynolds numbers, the resolution needs to be higher in the inlet region near the top of the step. This can be attributed to the singularity at that point. The higher Reynolds number cases also require increased resolution around the reattachment point of the main eddy.

Consider the grid system for the $\mathrm{Re} = 400$ case. The refined grid structure is fragmented, probably due to a noisy truncation error estimate coupled with the choice of $\tau^*$. Increasing the clustering distances $d_x$ and $d_y$ would result in some of the grid fragments coalescing. Despite the fragmentation, however, the solutions field is smooth and the calculated reattachment length agrees with accepted value. For the other cases examined, for both the driven cavity and backward-facing step problems, the overall structure of the grid system is not very dependent on the exact values of $d_x$ and $d_y$. For example, for the driven cavity problem at $\mathrm{Re} = 5000$ gridpoints are marked at each of the corners of the domain leading to the addition of subgrids there (see Fig. 5). For any of these subgrids to amalgamate with any other would require "large" values of $d_x$ and $d_y$ (approximately 25% of the domain length). Smaller values lead to an identical grid structure to that shown in the figure.

## 4. Discussion and Conclusions

The automatic adaptive refinement technique when used with multigrid shows itself to be a very efficient means of obtaining steady-state Navier–Stokes solutions. Considerable memory savings as well as a reduction in the total CPU time are achieved. The automatic refinement approach is based on estimates of the truncation error; therefore the solution error is monitored. Thus the refinement can continue until the solution error is reduced to a satisfactory level. Alternatively, a fixed number of refinement levels can be specified.

The extension to three dimensions is quite straightforward since the smoothers are based on the primitive-variable formulation of the Navier–Stokes equations. Another worthwhile extension would be to allow for arbitrary geometries through coordinate transformations. There are no apparent difficulties preventing this development.

## Acknowledgments

Mark Thompson was supported by a CSIRO Postdoctoral Fellowship during this work. The authors wish to thank Dr. S. C. Caruso who kindly provided a version of his adaptive grid code as a starting

point for this work. Discussions with Professor J. Oliger ware also very fruitful. This work was partially supported by the Office of Naval Research through the Center for Large Scale Scientific Computing at Stanford.

## REFERENCES

1. U. GHIA, K. N. GHIA, AND C. T. SHIN, *J. Comput. Phys.* **48**, 387 (1982).
2. L. FUCHS AND H. S. ZHAO, *Int. J. Numer. Methods Fluids* **4**, 539 (1984).
3. S. P. VANKA, *J. Comput. Phys.* **65**, 138 (1986).
4. S. P. VANKA, *Comput. Meth. Appl. Mech. Eng.*, in press.
5. S. P. VANKA, *in* Turbulent Shear Flows Conference, Cornell University, August 1985.
6. S. G. RUBIN AND P. K. KHOSLA, *Comput. Fluids* **9** 163 (1981).
7. A. BRANDT AND N. DINAR, in *Numerical Methods for Partial Differential Equations*, edited by S. Parter (Academic Press, New York/London, 1979).
8. M. BARCUS, M. PERIC AND G. SCHEUERER, "Berechnung Zweidimensionaler Strömungsprobleme mit Mehrgetterverfahren," Lehrstuhl für Strömungsmechanik, Universität Erlangen, Erlangen, West Germany, 1987.
9. H. L. STONE, *SIAM J. Numer Anal.* **5**, 530 (1968).
10. J. F. THOMPSON, *AIAA J.* **22**, No. 11, 1505 (1984).
11. D. A. ANDERSON, "Adaptive Grid Methods for Partial Differential Equations: Advances in Grid Generation," ASME Fluids Engineering Conference, Houston, 1983.
12. G. W. HEDSTROM AND G. H. RODRIGUE, in *Multigrid Methods*, Lecture Notes in Mathematics Vol. 960 (Springer-Verlag, New York/Berlin, 1982).
13. S. CARUSO, *Adaptive Grid Techniques for Fluid Flow Problems*, Ph.D. thesis, Thermosciences Division, Department of Mechanical Engineering, Stanford University, CA, 1985, (unpublished).
14. S. V. PATANKAR, *Numerical Heat Transfer and Fluid Flow* (McGraw–Hill, New York, 1980).
15. W. AUZINGER AND H. J. STETTER, in *Multigrid Methods*, Lecture Notes in Mathematics Vol. 960, (Springer-Verlag, New York/Berlin, 1982).
16. H. J. STETTER, *Numer. Math.* **29**, 425 (1978).
17. P. W. HEMKER, *in* Lecture Notes in Mathematics Vol. 1228, p. 150 (Springer-Verlag, New York/ Berlin, 1986).
18. S. V. PATANKAR, *Num. Heat Transfer* **4** 409 (1981).
19. D. N. ALLEN AND R. V. SOUTHWELL, *Q. J. Mech. Eng. Appl. Math.* **8** 129 (1955).
20. D. B. SPALDING, *Int. J. Num. Methods Eng.* **4** 551 (1972).
21. J. KIM AND P. MOIN, *J. Comput. Phys.* **59**, 308 (1985).
22. W. HACKBUSCH, *Rev. Roum. Math. Pures Appl.* **26**, 1319 (1981).
23. B. P. LEONARD, *J. Comput. Meth. Appl. Math. Eng.* **19**, 59 (1979).
24. R. AVVA, Thermosciences Division, Mechanical Engineering, Stanford University, Stanford, CA, private communication (1987).
25. A. BRANDT, *Math. Comput.* **31**, No. 138, 333 (1977).
26. W. HACKBUSCH AND U. TROTTENBERG, *in Proceedings, Multigrid Methods Conference*, Köln-Porz, 1981.
27. A. BRANDT, von Karman Institute, Lecture Series, 1984-04 (1984).
28. M. J. BERGER, "Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations," Ph.D. thesis, Department of Computer Science, Stanford University, CA, 1982 (unpublished).
29. M. J. BERGER, in *Adaptive Computational Methods for Partial Differential Equations*, edited by I. Babuska, J. Chandra, and J. E. Flaherty (SIAM, Philadelphia, 1983).
30. D. BAI, "Local Mesh Refinement Multilevel Techniques," Ph.D. thesis, Dept. of Applied Mathematics, Weizmann Institute, Israel 1984 (unpublished).
31. K. BECKER AND U. TROTTENBERG, Paper No. 111., Gesellschaft für Mathematik Und Datenverarbeitung, MBH, Bonn, September 1984 (unpublished).

32. A. S. BENJAMIN AND V. E. DENNY, *J. Comput. Phys.* **33**, 340 (1979).
33. R. SCHREIBER AND H. B. KELLER, *J. Comput. Phys.* **49**, 310 (1983).
34. R. MEI AND A. PLOTKIN, *Comput. Fluids* **14**, No. 3, 239 (1986).
35. C. J. FREITAS, R. L. STREET, A. N. FINDIKAKIS, AND J. R. KOSEFF, *Int. J. Num. Meth. Fluids.* **5**, 561 (1985).
36. L. P. HACKMAN, G. D. RAITHBY, AND A. B. STRONG, *Int. J. Num. Meth. Fluids.* **4** 711 (1984).
37. H. S. RHEE, J. R. KOSEFF, AND R. L. STREET, *Exp. Fluids* **2**, 57 (1984).
38. B. F. ARMALY, F. DURST, J. C. F. PEREIRA, AND B. SCHONUNG, *J. Fluid. Mech.* **127**, 473 (1983).