INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 13, 971–997 (1991)

NUMERICAL COMPUTATION OF UNSTEADY INCOMPRESSIBLE FLOW IN COMPLEX GEOMETRY USING A COMPOSITE MULTIGRID TECHNIQUE

M. HINATSU

Ship Performance Division, Ship Research Institute, Ministry of Transport of Japan, 6-38-1, Shinkawa, Mitaka, Tokyo, 181, Japan

AND

J. H. FERZIGER

Thermosciences Division, Department of Mechanical Engineering, Stanford University, Stanford, CA 94305, U.S.A.

SUMMARY

This paper presents a composite multigrid method and its application to a geometrically complex flow. The treatment of the interior boundary conditions within a composite multigrid strategy is described in detail for a 1D model equation. For the Navier–Stokes equations a staggered grid technique is adopted for spatial discretization and a fractional step method is used for the time advance. Lid-driven cavity flows are used to demonstrate the effectiveness of the method.

KEY WORDS Unsteady Navier–Stokes Composite multigrid method Fractional step method Staggered grid

1. INTRODUCTION

Many methods for numerical simulation of fluid flows have been proposed and CFD (computational fluid dynamics) has become a practical design tool. However, problems remain with regard to computational efficiency, accuracy and turbulence modelling. The most difficult problem may be mesh generation for geometrically complicated domains.

Many schemes have been devised to cope with these issues. The multigrid method[1] is one of the most efficient schemes for elliptic problems and has been applied to the Navier–Stokes equations. To deal with geometric complexity, domain decomposition has been proposed; its origins go back to Schwarz.[2] The idea is that a complicated domain be decomposed into subdomains whose geometry is simple enough to be easily gridded.

In the present work we apply a combination of these two methods, i.e. a composite multigrid strategy, to flows in geometrically complicated domains. In particular we treat 2D unsteady laminar flows. The Navier–Stokes equations are discretized using second-order central differencing on a staggered grid in space and a fractional step time advance method. The velocity components are advanced explicitly and the pressure is obtained by solving a Poisson equation using a composite multigrid method.[3] The momentum equations are integrated independently on each subgrid. Interpolation on the composite grid is accomplished with a Coons patch method.[4]

0271–2091/91/180971–27$13.50

Received 16 August 1990
Revised 16 January 1991

Prior to solving the Navier–Stokes equations, we investigate the effectiveness of the composite multigrid technique for a 1D model equation. We also discuss the interior boundary conditions.

After demonstrating the properties of the composite multigrid method, we apply it to the Navier–Stokes equations. First we check the accuracy of the method for lid-driven cavity flow at $Re=3200$ and evaluate the error. Then we simulate geometrically complex cavity flows.

2. THE GOVERNING EQUATIONS

The 2D unsteady incompressible Navier–Stokes (N–S) equations can be written in the following form in a curvilinear co-ordinate system:

$$\frac{\partial \hat{\mathbf{q}}}{\partial t}+\frac{\partial \mathbf{E}}{\partial \xi}+\frac{\partial \mathbf{F}}{\partial \eta}+\frac{\partial \mathbf{H}_\xi}{\partial \xi}+\frac{\partial \mathbf{H}_\eta}{\partial \eta}+\frac{\partial \mathbf{P}}{\partial \xi}+\frac{\partial \mathbf{Q}}{\partial \eta}=0, \tag{1}$$

where

$$\hat{\mathbf{q}}=\mathbf{q}/J, \qquad \mathbf{q}=[u,v]^{\mathrm{T}}, \qquad J=\xi_x\eta_y-\xi_y\eta_x, \tag{2}$$

u and v are the Cartesian velocity components along the x- and y-axes and ξ and η are arbitrary curvilinear co-ordinates. J is the Jacobian. Subscripts x and y stand for derivatives with respect to x and y. The second and third terms of the LHS of (1) are the convection terms, so that $\mathbf{E}$ and $\mathbf{F}$ are

$$\mathbf{E}=U\hat{\mathbf{q}}, \qquad \mathbf{F}=V\hat{\mathbf{q}}, \tag{3a}$$

where

$$U=u\xi_x+v\xi_y, \qquad V=u\eta_x+v\eta_y \tag{3b}$$

are the contravariant velocity components; however, we shall use the Cartesian velocity components as the primary variables. The fourth and fifth terms are the pressure gradient terms; $\mathbf{H}_\xi$ and $\mathbf{H}_\eta$ can be written as

$$\mathbf{H}_\xi=\begin{pmatrix} H_{\xi u}\\ H_{\xi v}\end{pmatrix}=\begin{pmatrix}\xi_x p/J\\ \xi_y p/J\end{pmatrix}, \qquad \mathbf{H}_\eta=\begin{pmatrix} H_{\eta u}\\ H_{\eta v}\end{pmatrix}=\begin{pmatrix}\eta_x p/J\\ \eta_y p/J\end{pmatrix}, \tag{4}$$

where p is the pressure. The last two terms are the viscous terms; $\mathbf{P}$ and $\mathbf{Q}$ have the form

$$\mathbf{P}=\frac{1}{J\,Re}\begin{pmatrix}(\xi_x^2+\xi_y^2)u_\xi+(\xi_x\eta_x+\xi_y\eta_y)u_\eta\\ (\xi_x^2+\xi_y^2)v_\xi+(\xi_x\eta_x+\xi_y\eta_y)v_\eta\end{pmatrix},$$
$$\mathbf{Q}=\frac{1}{J\,Re}\begin{pmatrix}(\eta_x^2+\eta_y^2)u_\eta+(\xi_x\eta_x+\xi_y\eta_y)u_\xi\\ (\eta_x^2+\eta_y^2)v_\eta+(\xi_x\eta_x+\xi_y\eta_y)v_\xi\end{pmatrix}, \tag{5}$$

where Re is the Reynolds number. The above equations have to be solved simultaneously with the continuity equation

$$\frac{1}{J}\left[\frac{\partial}{\partial \xi}\left(\xi_x\frac{u}{J}+\xi_y\frac{v}{J}\right)+\frac{\partial}{\partial \eta}\left(\eta_x\frac{u}{J}+\eta_y\frac{v}{J}\right)\right]=\frac{1}{J}\left[\frac{\partial}{\partial \xi}\left(\frac{U}{J}\right)+\frac{\partial}{\partial \eta}\left(\frac{V}{J}\right)\right]=0. \tag{6}$$

3. NUMERICAL SCHEME

3.1. Differencing in space and time

The momentum equations for the velocity components u and v can be written as

$$\frac{\partial}{\partial t}\left(\frac{u}{J}\right)+L^u(u, v, u_\xi, u_\eta, v_\xi, v_\eta, p)=0, \tag{7}$$

$$\frac{\partial}{\partial t}\left(\frac{v}{J}\right)+L^v(u, v, u_\xi, u_\eta, v_\xi, v_\eta, p)=0, \tag{8}$$

where L^u and L^v are the sum of the convection, pressure gradient and viscous diffusion operators. Multiplying (7) by ξ_x or η_x and multiplying (8) by ξ_y or η_y and summing, we get the equation for U and V:

$$\frac{\partial}{\partial t}\frac{1}{J}\begin{pmatrix}U\\ V\end{pmatrix}+\begin{pmatrix}\xi_x\\ \eta_x\end{pmatrix}L^u(u, v, \ldots, p)+\begin{pmatrix}\xi_y\\ \eta_y\end{pmatrix}L^v(u, v, \ldots, p)=0. \tag{9}$$

To discretize this equation in space, we use second-order central differencing on a staggered grid in which the pressure node is located at the cell centre and the contravariant velocity components U and V are located on the cell boundaries as shown in Figure 1. Control volume I is used for the x-momentum equation and control volume II for the y-momentum equation. In order to evaluate u and v at the velocity nodes, we must evaluate U and V at those nodes, but on a staggered grid U and V are known at different points. Maliska and Raithby[5] used the average of the values at four surrounding points to obtain the velocity component on a node of the other component. For example, to evaluate V at the point A of Figure 1, they took an average of V at B, C, D and E. We adopt this method.

After evaluating U and V, we get u and v from the following relation:

$$\begin{pmatrix}u\\ v\end{pmatrix}=\begin{pmatrix}\xi_x & \xi_y\\ \eta_x & \eta_y\end{pmatrix}^{-1}\begin{pmatrix}U\\ V\end{pmatrix}. \tag{10}$$

For discretization of the convection terms we use the QUICK scheme.[6]

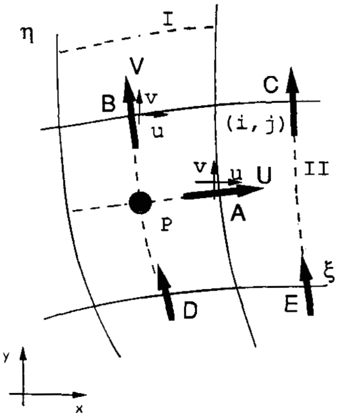


Figure 1. Staggered grid system for velocity and pressure

Time advance is carried out using the fractional step method. Let u^n, v^n and p^n be the velocities and pressure at time step n and assume that u^n and v^n satisfy the continuity equation.

First we evaluate $U^{\overline{n+1}}$ and $V^{\overline{n+1}}$ by explicit integration of (9):

$$\frac{1}{J}\begin{pmatrix}U\\V\end{pmatrix}^{\overline{n+1}}=\frac{1}{J}\begin{pmatrix}U\\V\end{pmatrix}^{n}-\Delta t\left[L^u(u^n,v^n,\ldots,p^n)\begin{pmatrix}\xi_x\\\eta_x\end{pmatrix}+L^v(u^n,v^n,\ldots,p^n)\begin{pmatrix}\xi_y\\\eta_y\end{pmatrix}\right]. \tag{11}$$

This is the predictor of the MacCormack scheme[7] and many others.

The corrector for the MacCormack scheme is

$$\begin{aligned}\frac{1}{J}\begin{pmatrix}\tilde U\\\tilde V\end{pmatrix}=\frac{1}{2}\Bigg\{\frac{1}{J}\begin{pmatrix}U\\V\end{pmatrix}^{n}+\frac{1}{J}\begin{pmatrix}U\\V\end{pmatrix}^{\overline{n+1}}&-\Delta t\Bigg[L^u(u^{\overline{n+1}},v^{\overline{n+1}},\ldots,p^n)\begin{pmatrix}\xi_x\\\eta_x\end{pmatrix}\\&+L^v(u^{\overline{n+1}},v^{\overline{n+1}},\ldots,p^n)\begin{pmatrix}\xi_y\\\eta_y\end{pmatrix}\Bigg]\Bigg\},\end{aligned} \tag{12}$$

where $u^{\overline{n+1}}$, $v^{\overline{n+1}}$ are computed from $U^{\overline{n+1}}$, $V^{\overline{n+1}}$ using (10).

Using (11), equation (12) can be rewritten as

$$\begin{aligned}\frac{1}{J}\begin{pmatrix}\tilde U\\\tilde V\end{pmatrix}-\frac{1}{J}\begin{pmatrix}U\\V\end{pmatrix}^{n}=-\frac{1}{2}\Delta t\Bigg[&L^u(u^n,v^n,\ldots,p^n)\begin{pmatrix}\xi_x\\\eta_x\end{pmatrix}+L^v(u^n,v^n,\ldots,p^n)\begin{pmatrix}\xi_y\\\eta_y\end{pmatrix}\\&+L^u(u^{\overline{n+1}},v^{\overline{n+1}},\ldots,p^n)\begin{pmatrix}\xi_x\\\eta_x\end{pmatrix}+L^v(u^{\overline{n+1}},v^{\overline{n+1}},\ldots,p^n)\begin{pmatrix}\xi_y\\\eta_y\end{pmatrix}\Bigg].\end{aligned} \tag{13}$$

The new velocity field $\tilde u$, $\tilde v$ does not satisfy the continuity equation (10) so we introduce a pressure correction and compute the new velocity field U^{n+1}, V^{n+1} and u^{n+1}, v^{n+1} via (10) from

$$\frac{1}{J}\begin{pmatrix}U^{n+1}\\V^{n+1}\end{pmatrix}-\frac{1}{J}\begin{pmatrix}\tilde U\\\tilde V\end{pmatrix}=-\left\{\left(\frac{\partial}{\partial\xi}(\Delta H_{\xi u})+\frac{\partial}{\partial\eta}(\Delta H_{\eta u})\right)\begin{pmatrix}\xi_x\\\eta_x\end{pmatrix}-\left(\frac{\partial}{\partial\xi}(\Delta H_{\xi v})+\frac{\partial}{\partial\eta}(\Delta H_{\eta v})\right)\begin{pmatrix}\xi_y\\\eta_y\end{pmatrix}\right\}\Delta t, \tag{14}$$

where $\Delta H_{\xi u}=H_{\xi u}^{n+1}-H_{\xi u}^{n}$, etc. are determined by forcing U^{n+1} and V^{n+1} to satisfy the continuity equation. In this way we obtain a pressure equation (strictly speaking, a pressure increment equation)

$$\begin{aligned}&\frac{\partial}{\partial\xi}\left[\xi_x\left(\frac{\partial}{\partial\xi}(\Delta H_{\xi u})+\frac{\partial}{\partial\eta}(\Delta H_{\eta u})\right)+\xi_y\left(\frac{\partial}{\partial\xi}(\Delta H_{\xi v})+\frac{\partial}{\partial\eta}(\Delta H_{\eta v})\right)\right]\\&\quad+\frac{\partial}{\partial\eta}\left[\eta_x\left(\frac{\partial}{\partial\xi}(\Delta H_{\xi u})+\frac{\partial}{\partial\eta}(\Delta H_{\eta u})\right)+\eta_y\left(\frac{\partial}{\partial\xi}(\Delta H_{\xi v})+\frac{\partial}{\partial\eta}(\Delta H_{\eta v})\right)\right]=\left\{\frac{\partial}{\partial\xi}\left(\frac{\tilde U}{J}\right)+\frac{\partial}{\partial\eta}\left(\frac{\tilde V}{J}\right)\right\}\frac{1}{\Delta t}\end{aligned} \tag{15}$$

This is a Poisson equation for $\Delta p=p^{n+1}-p^n$ and can be solved by the multigrid method. On a Cartesian staggered grid no boundary condition is needed for the pressure; in a curvilinear co-ordinate system a pressure boundary condition is usually needed. However, if in the latter case the co-ordinate system is orthogonal at the boundary, the need for a boundary condition disappears. In this work we require grid orthogonality at the boundary so no boundary condition is needed. However, the discrete system of equations is singular. For a simplicity of coding we use a fictitious pressure node outside the computational domain to eliminate the singularity. Furthermore, to ensure uniqueness, we set the solution at an arbitrary point (e.g. $(\xi,\eta)=(1/2,1/2)$) to zero. Other methods of desingularizing the system are available.

3.2. Multigrid method for the pressure

Multigrid[1] is a well-known method for solving elliptic equations. We apply it only to the pressure equation. The essential idea of the multigrid method is reduction of the error using several grids of different sizes; high-frequency components of the error are removed by smoothing on a fine grid and low-frequency components are damped on coarser grids.

The system of linear equations (15) can be written

$$L\phi = f, \tag{16}$$

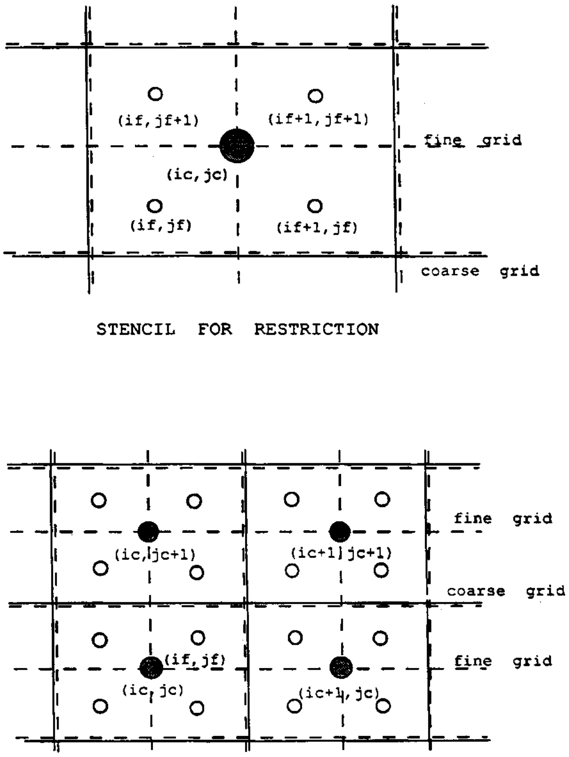
where ϕ is the solution and f is the source (forcing) term.

After we relax (16) using a suitable smoother on the finest grid, we get an approximate solution $\tilde{\phi}^k$ and

$$L^k\tilde{\phi}^k = f - R^k, \tag{17}$$

where R is the residual and superscript k stands for the grid level; $k = k_{\max}$ corresponds to the finest grid and $k-1$ to a grid twice as coarse. The smoother should reduce the high-frequency components of the error rapidly. Subtracting (17) from (16), we get an equation for the error $\tilde{e}^k$:

$$L^k\tilde{e}^k = R^k, \qquad \tilde{e}^k = \phi - \tilde{\phi}^k. \tag{18}$$



STENCIL FOR RESTRICTION

STENCIL FOR PROLONGATION

Figure 2. Stencils of the restriction and prolongation operators used in the multigrid method

Next we remove lower-frequency components of $\tilde{e}$. To do that, (18) is solved on grid $k-1$:

$$L^{k-1}\tilde{e}^{k-1}=R^{k-1}, \qquad R^{k-1}=I_k^{k-1}R^k, \tag{19}$$

where I_k^{k-1} is a restriction operator. After smoothing $\tilde{e}$ on the coarse grid, $\tilde{e}$ is interpolated onto the fine grid by interpolation or prolongation:

$$\tilde{e}^k=I_{k-1}^k\tilde{e}^{k-1}, \tag{20}$$

where I_{k-1}^k is an interpolation operator. The solution is then updated:

$$\phi^k=\phi_{\text{old}}^k+\tilde{e}^k. \tag{21}$$

In the present work, since we use a staggered grid system, I_k^{k-1} and I_{k-1}^k are defined as follows:

$$I_k^{k-1}\phi_{ic,jc}^k=(\phi_{if,jf}^k+\phi_{if+1,jf}^k+\phi_{if,jf+1}^k+\phi_{if+1,jf+1}^k)/4, \tag{22}$$

where $if=2ic-1$ and $jf=2jc-1$,

$$I_{k-1}^k\phi_{if,jf}^k=(9\phi_{ic,jc}^k+3\phi_{ic+1,jc}^k+3\phi_{ic,jc+1}^k+\phi_{ic+1,jc+1}^k)/16, \tag{23}$$

where $ic=1+if/2$ and $jc=1+jf/2$ (see Figure 2).

The method is readily extended to larger numbers of grids.

4. DOMAIN DECOMPOSITION TECHNIQUE

In order to solve the Navier–Stokes equations, we have to divide the domain using a suitable grid system. For flow fields that do not have simple geometry, covering the entire domain with a single grid is difficult.

Domain decomposition is a method of coping with this problem. In this technique a geometrically complicated domain is divided into several simpler ones. Schwarz[2] proposed an alternating solution procedure for elliptic problems. Since this method uses Dirichlet conditions on the interior boundaries, it requires that the subdomains overlap.

Van der Wijngaart[8] revised the Schwarz method using asymmetric interior boundary conditions. If Neumann conditions are used for the interior boundary condition of grid I, Dirichlet conditions are applied on grid II. He showed that this treatment allows the requirement of subdomain overlap to be removed. More recently, Lions[9] developed a theory of the Schwarz method on non-overlapping subdomains.

A few applications of this method to fluid flow simulation exist. Meakin and Street[10] simulated a 3D environmental flow using a composite grid method. Van der Wijngaart[8] developed the SWAPR (Schwarz alternating procedure revised) method which uses asymmetric boundary conditions on the interior boundary.

In the present work we combine domain decomposition with the multigrid method. Henshaw and Chesshire[11] solved the Poisson equation using a composite multigrid technique. Perng[12] and Perng and Street[3] simulated flows in complicated domains using a combination of explicit time advance on individual grids and a multigrid pressure solver on the complete composite grid. They showed the effectiveness of the composite multigrid method for geometrically complicated flows including 3D problems. However, they restricted the domain decomposition by requiring that grid nodes in overlapping domains belong to both grids. This restriction reduces freedom in grid construction and is removed in the present work. On the interior boundaries Perng and Street adopted Neumann boundary conditions but also used Dirichlet conditions.

Before describing the Navier–Stokes solver, we consider the composite multigrid method for a simple 1D model problem.

4.1. Multigrid composite grid technique (1D model problem)

Consider the following 1D model problem:

$$y''=2 \quad (0\leqslant x\leqslant 1), \quad \text{with } y(0)=0,\ y'(1)=1. \tag{24}$$

The exact solution is

$$y=x(x-1). \tag{25}$$

We choose two overlapping grids, one from $x=0$ to $x=0{\cdot}6$ and the other covering $x=0{\cdot}4$ to $x=1$; we call the former grid I and the latter grid II. Each grid is divided into 16 equal elements, so the finest grid size is 0·0375. Three levels are used on each grid; the finest has 16 intervals and the coarsest four intervals. In this 1D problem we adopt a regular grid, so the coarse grid nodes coincide with the fine grid nodes.

To use Schwarz iteration, we have to estimate the interior boundary value by interpolating from the other grid. Two solution methods are considered. In the first, interior boundary value communication is limited to the finest grid and the coarse grid smoothings are carried out independently. In the second method, data communication at the interior boundary is allowed at every grid level. Hereafter we call the former the incomplete composite multigrid (or simply ICMG) method and the latter the complete composite multigrid (or CCMG) method. Henshaw and Chesshire[11] used the CCMG method while Perng and Street[12] used the ICMG method.

The algorithms for the two methods are as follows.

ICMG method

(i) Iterate the equation on grid I using a local V-cycle with a guessed interior boundary value.
(ii) The interior boundary value for the finest level of grid II is obtained by interpolation on grid I.
(iii) Iterate the equation on grid II using a local V-cycle.
(iv) Find the boundary condition at the finest level of grid I by interpolation on grid II.
(v) Repeat steps (i)–(iv) until the solution converges.

CCMG method

(i) Iterate the equation on the finest level of grid I using a guessed interior boundary value.
(ii) The interior boundary condition for the finest level of grid II is found by interpolation on grid I.
(iii) Iterate the equation on the finest level of grid II.
(iv) Interpolate to find the grid I boundary condition.
(v) Repeat steps (i)–(iv) (Schwarz iteration on the finest grid level). There is no need to iterate to convergence. In the test computation we iterated twice.
(vi) The residual on each composite grid is restricted to the coarser grid and the correction is iterated using procedures (i)–(v).
(vii) After solving at the coarsest grid level, work back to the finer grids.
(viii) Iterate procedures (i)–(vii) (V-cycle) until the solution converges.

4.2. Interior boundary condition for multigrid composite grid technique

At the interior boundaries we tried two types of conditions, Neumann and Dirichlet.

First consider the ICMG method. In this case data communication is limited to the finest grid and the interior boundary condition is found as follows.

For the Dirichlet boundary condition case,

$$y^{\mathrm{I}}_{\mathrm{right}}=\mathrm{ITP}(y^{\mathrm{II}}), \qquad y^{\mathrm{II}}_{\mathrm{left}}=\mathrm{ITP}(y^{\mathrm{I}}), \tag{26}$$

where ITP is an interpolation operator, y^{I} means the solution on grid I and $y^{\mathrm{I}}_{\mathrm{right}}$ is the right boundary value on grid I.

We can also introduce an overrelaxation parameter to accelerate convergence as was done by Tang:[13]

$$y^{\mathrm{I}}_{\mathrm{right}}=\omega\mathrm{ITP}(y^{\mathrm{II}})+(1-\omega)(y^{\mathrm{I}}_{\mathrm{right}})_{\mathrm{old}}. \tag{27}$$

In a similar manner we find $y^{\mathrm{II}}_{\mathrm{left}}$.

The case of Neumann boundary conditions is similar:

$$\mathrm{d}y^{\mathrm{I}}_{\mathrm{right}}/\mathrm{d}x=\mathrm{ITP}(\mathrm{d}y^{\mathrm{II}}/\mathrm{d}x)$$

or

$$\mathrm{d}y^{\mathrm{I}}_{\mathrm{right}}/dx=\omega\mathrm{ITP}(\mathrm{d}y^{\mathrm{II}}/\mathrm{d}x)+(1-\omega)(\mathrm{d}y^{\mathrm{I}}_{\mathrm{right}}/\mathrm{d}x)_{\mathrm{old}}. \tag{28}$$

In the ICMG method, relaxation on each set of coarse grids is carried out independently; thus the interior boundary condition for the error is

$$\begin{aligned}\tilde{e}^{\mathrm{I}}_{\mathrm{right}}&=0 \quad \text{(Dirichlet)},\\ \mathrm{d}\tilde{e}^{\mathrm{I}}_{\mathrm{right}}/\mathrm{d}x&=0 \quad \text{(Neumann)}.\end{aligned} \tag{29}$$

Next consider the CCMG method. The boundary condition on the finest grid level is found as in the ICMG case. On the coarser grids the treatment of the boundary condition is somewhat more complicated.

After iterating the correction equation on the coarse grid, the result satisfies the following relation:

$$\begin{aligned}(y^{\mathrm{I}}_{\mathrm{new}})_{\mathrm{right}}&=\mathrm{ITP}(y^{\mathrm{II}}_{\mathrm{new}}) && \text{(Dirichlet)},\\ (\mathrm{d}y^{\mathrm{I}}_{\mathrm{new}}/\mathrm{d}x)_{\mathrm{right}}&=\mathrm{ITP}(\mathrm{d}y^{\mathrm{II}}_{\mathrm{new}}/\mathrm{d}x) && \text{(Neumann)}.\end{aligned}$$

In other words, instead of (29), the boundary condition on the error on the interior boundary should be

$$\begin{aligned}(y^{\mathrm{I}}_{\mathrm{old}}+\tilde{e}^{\mathrm{I}})_{\mathrm{right}}&=\mathrm{ITP}(y^{\mathrm{II}}_{\mathrm{old}})+\mathrm{ITP}(\tilde{e}^{\mathrm{II}}) && \text{(Dirichlet)},\\ (\mathrm{d}y^{\mathrm{I}}_{\mathrm{old}}/\mathrm{d}x+\mathrm{d}\tilde{e}^{\mathrm{I}}/\mathrm{d}x)_{\mathrm{right}}&=\mathrm{ITP}(\mathrm{d}y^{\mathrm{II}}_{\mathrm{old}}/\mathrm{d}x)+\mathrm{ITP}(\mathrm{d}\tilde{e}^{\mathrm{II}}/\mathrm{d}x) && \text{(Neumann)}.\end{aligned} \tag{30}$$

Again we can accelerate the boundary conditions:

$$\tilde{e}^{\mathrm{I}}=\omega\tilde{e}^{*}+(1-\omega)\tilde{e}^{\mathrm{I}}_{\mathrm{old}},$$

where $\tilde{e}^{*}=\mathrm{ITP}(y^{\mathrm{II}}_{\mathrm{old}})+\mathrm{ITP}(\tilde{e}^{\mathrm{II}})-y^{\mathrm{I}}_{\mathrm{old}}$. The Neumann boundary condition can be treated in a similar manner.

Although (30) is the exact boundary condition for the error, we may approximate it in a way that makes data communication at each grid level independent, namely

$$\tilde{e}^{\mathrm{I}}_{\mathrm{right}}=\mathrm{ITP}(\tilde{e}^{\mathrm{II}}) \quad \text{or} \quad \mathrm{d}\tilde{e}^{\mathrm{I}}_{\mathrm{right}}/\mathrm{d}x=\mathrm{ITP}(\mathrm{d}\tilde{e}^{\mathrm{II}}/\mathrm{d}x). \tag{31}$$

In the present study we tested both (30) and (31).

4.3. Remarks on the 1D model problem

Next we describe some 1D test computations. The relaxation method was Gauss–Seidel. Two sweeps were performed at each grid level. On the coarsest grid, iteration was carried to convergence. For the ICMG method, one V-cycle was made on each subgrid and the total number of Schwarz iterations was limited to 20. For the CCMG method, two Schwarz iterations were performed at each grid level and 10 V-cycles were carried out. Thus, the work is almost the same for the two methods. Because the CCMG method requires data communication on each grid level, its cost is slightly larger.

Now we present results for the 1D test problem. The test computations are (a) the ICMG method with Dirichlet interior boundary conditions (IBCs), (b) the ICMG method with Neumann IBCs, (c) the CCMG method with Dirichlet IBCs (equation (30)), (d) the CCMG method with Dirichlet IBCs (equation (31)), (e) the CCMG method with Neumann IBCs (equation (30)) and (f) the CCMG method with Neumann IBCs (equation (31)). We show the normalized errors after 20 Schwarz (one V-cycle) iterations for the ICMG method and after two Schwarz (10 V-cycles) iterations for the CCMG method to judge the rapidity of convergence. The

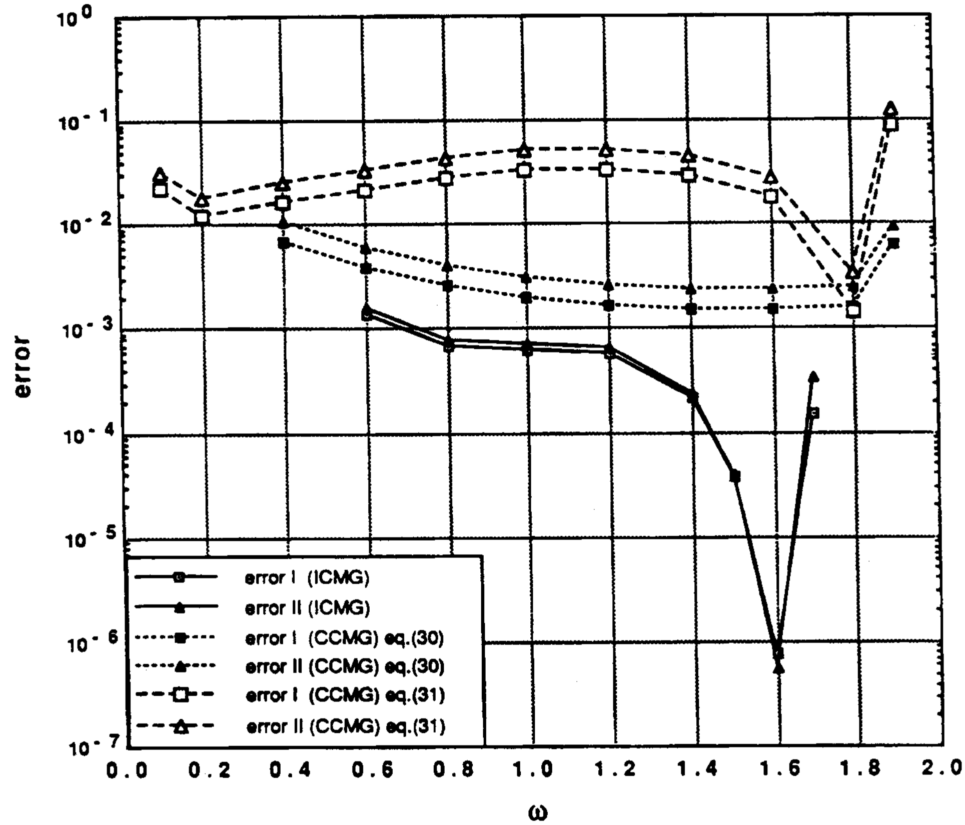


Figure 3. Comparison of errors in the ICMG and CCMG methods after 20 Schwarz one-V-cycle iterations for the ICMG method and two Schwarz 10-V-cycle iterations for the CCMG method with Dirichlet IBCs and Neumann–Dirichlet boundary conditions at the domain boundary

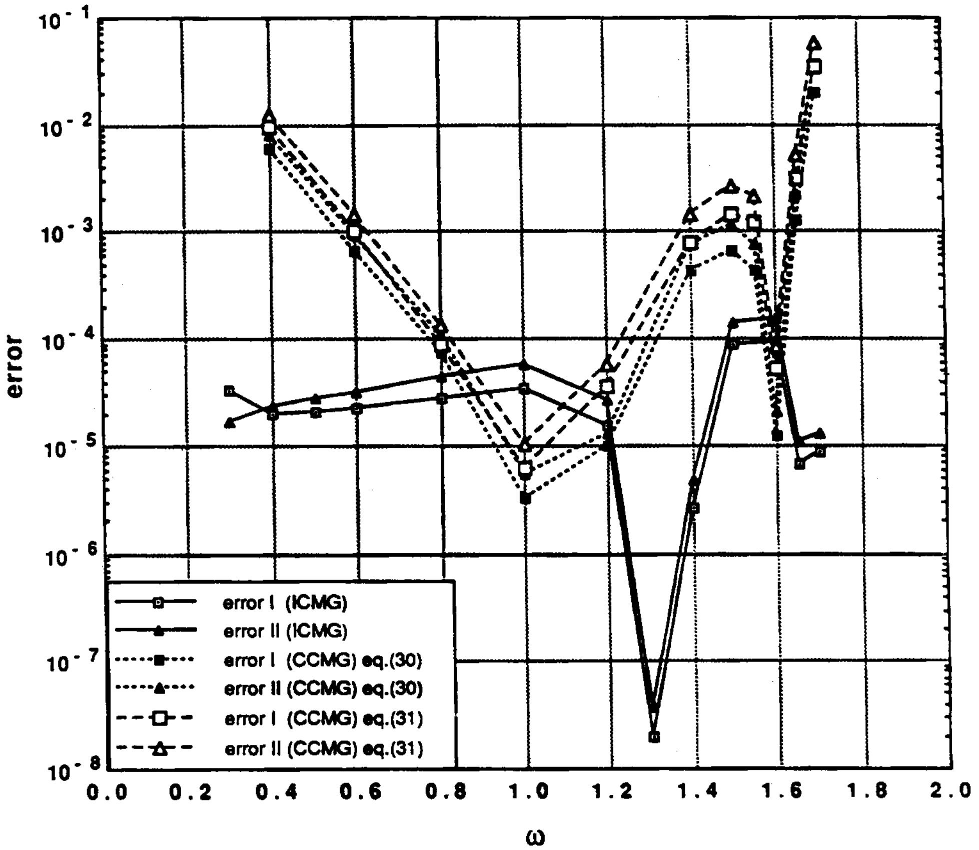


Figure 4. Comparison of errors in the ICMG and CCMG methods after 20 Schwarz one-V-cycle iterations for the ICMG method and two Schwarz 10-V-cycle iterations for the CCMG method with Neumann IBCs and Neumann–Dirichlet boundary conditions at the domain boundary

error is defined by

$$\text{error}=\sqrt{\left(\frac{\sum(y_{\text{numerical}}-y_{\text{exact}})^2}{\sum y_{\text{exact}}^2}\right)}. \tag{32}$$

The errors are given as functions of the acceleration parameter ω. Figure 3 shows the errors with Dirichlet IBCs for the ICMG and CCMG methods and Figure 4 gives the errors with Neumann IBCs for the same methods. Error I is the error on the finest level of subgrid I. In both figures, results obtained with both (30) and (31) are shown for the CCMG method. The ICMG method converges faster with Neumann IBCs than with Dirichlet IBCs. In both cases the convergence is accelerated by overrelaxation of the interior boundary values. The optimal relaxation parameters are $\omega\simeq 1{\cdot}6$ in the Dirichlet IBC case and $\omega\simeq 1{\cdot}3$ in the Neumann IBC case.

The ICMG method converges faster than the CCMG method, especially with Dirichlet IBCs. For both types of IBCs, (30) gives faster convergence than does (31); the difference in CPU time per time step is very slight, so the method based on (30) is faster overall. Interior boundary value acceleration is less effective in the CCMG method. With Neumann IBCs the best convergence in the CCMG method is obtained with $\omega=1$.

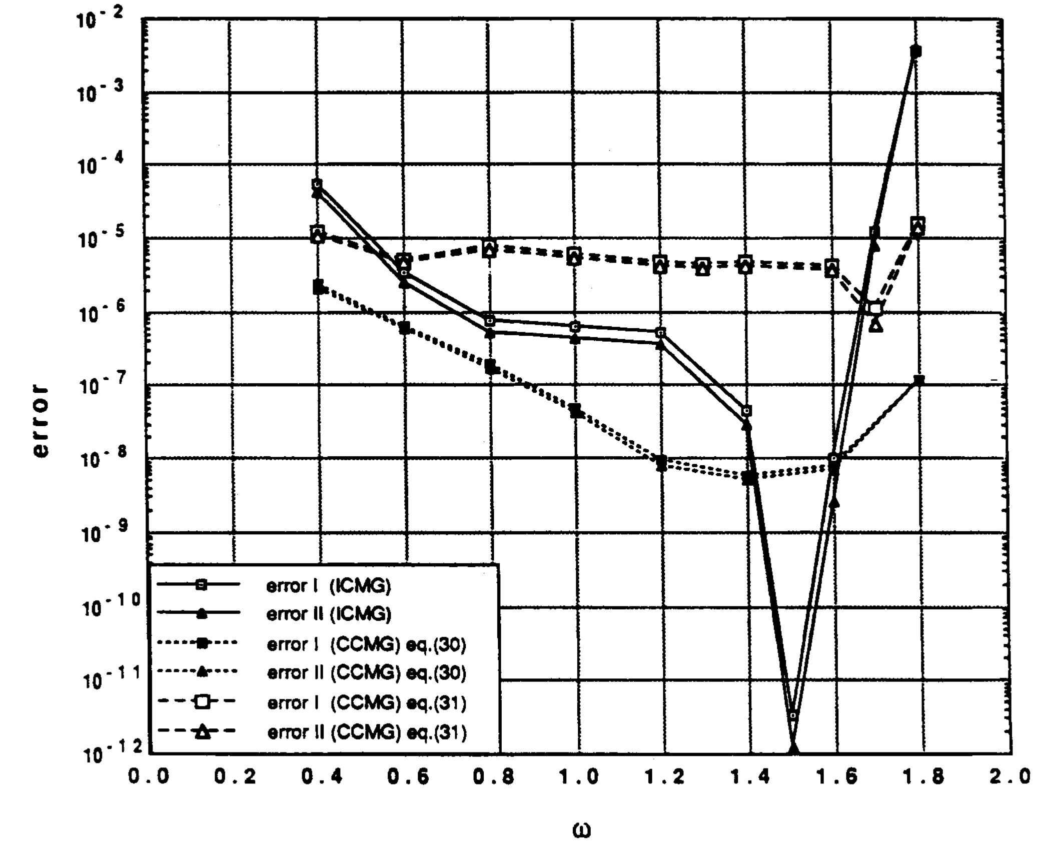


Figure 5. Comparison of errors in the ICMG and CCMG methods after 20 Schwarz one-V-cycle iterations for the ICMG method and two Schwarz 10-V-cycle iterations for the CCMG method with Dirichlet IBCs and Dirichlet boundary conditions at the domain boundary

However, these trends change with the external boundary conditions at $y=0$ and 1. For instance, when Dirichlet boundary conditions are given, i.e. $y(0)=y(1)=0$, with Dirichlet IBCs, the error as a function of ω is shown in Figure 5. The CCMG method converges much faster than in the previous case. However, for optimal ω (in this case $\omega=1{\cdot}5$) the ICMG method remains faster.

The convergence rate also depends on the size of the overlap region. For these tests the external boundary conditions were $y(0)=0$, $y'(1)=1$ and Neumann IBCs were used. Test computations were made with 20%, 30%, 40% and 50% overlap. Figure 6 shows the error for the ICMG method after 20 Schwarz iterations as a function of the overlap. The number of grid points and the number of levels are the same as before, i.e. 16 equal elements and three multigrid levels. The optimal acceleration parameter is only weakly affected by the amount of overlap, but the convergence is much faster when the overlap is large.

Finally we compared the convergence speed between a single (non-composite) grid and a composite grid. We solved (24) using a 32-point single grid and a 50% overlap composite grid. The finest level of the composite grid has 24 intervals of the same size as the finest non-composite grid. Three levels were used in both cases. The ICMG method was used on the composite grid

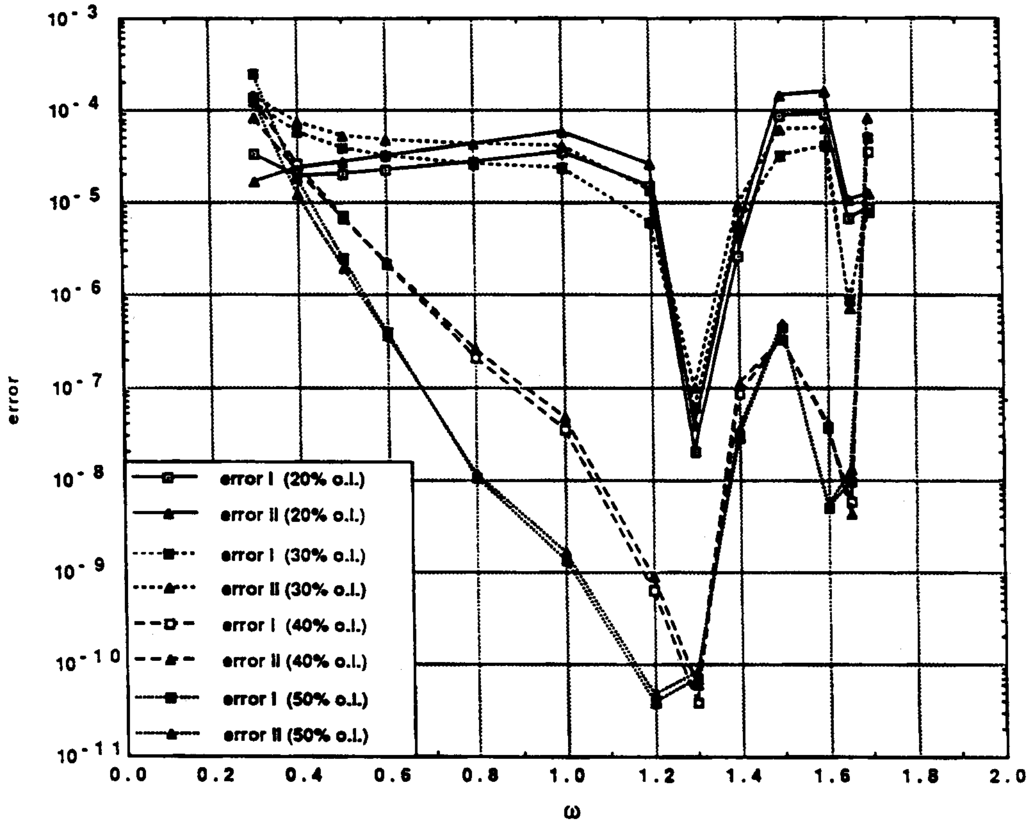


Figure 6. Dependence of convergence rate on the size of the overlap region for the ICMG method

with $\omega = 1{\cdot}2$. Figure 7 shows the maximum residual on each subgrid. Since we must guess the initial IBCs for the ICMG method, the residual at the first iteration is much bigger than in the non-composite grid method. This requires the ICMG method to take more iterations. However, the convergence rate is almost same, so the cost increase is almost entirely due to the overlap.

4.4. Composite multigrid method for a 2D model problem

The method can be applied to 2D boundary value problems. Since we intend to apply the method to the pressure equation, the Poisson equation was chosen as a test problem. In the test computation the domain consisted of two squares shifted diagonally by 40% of the diagonal length. The physical domain and the mesh are shown in Figure 8.

Neumann boundary conditions were applied at the entire boundary of the computational domain. We introduced four singularities as the forcing; their strengths add to zero for consistency with the boundary condition $\partial\phi/\partial n = 0$:

$$\int_S \nabla^2\phi \, \mathrm{d}S = \int_S f \, \mathrm{d}S = \int_L \frac{\partial\phi}{\partial n} \, \mathrm{d}l = 0. \tag{33}$$

For the interior boundary condition we can use Dirichlet or Neumann conditions. Because the fine grid nodes do not coincide with nodes of the coarse grid, if we use Dirichlet boundary

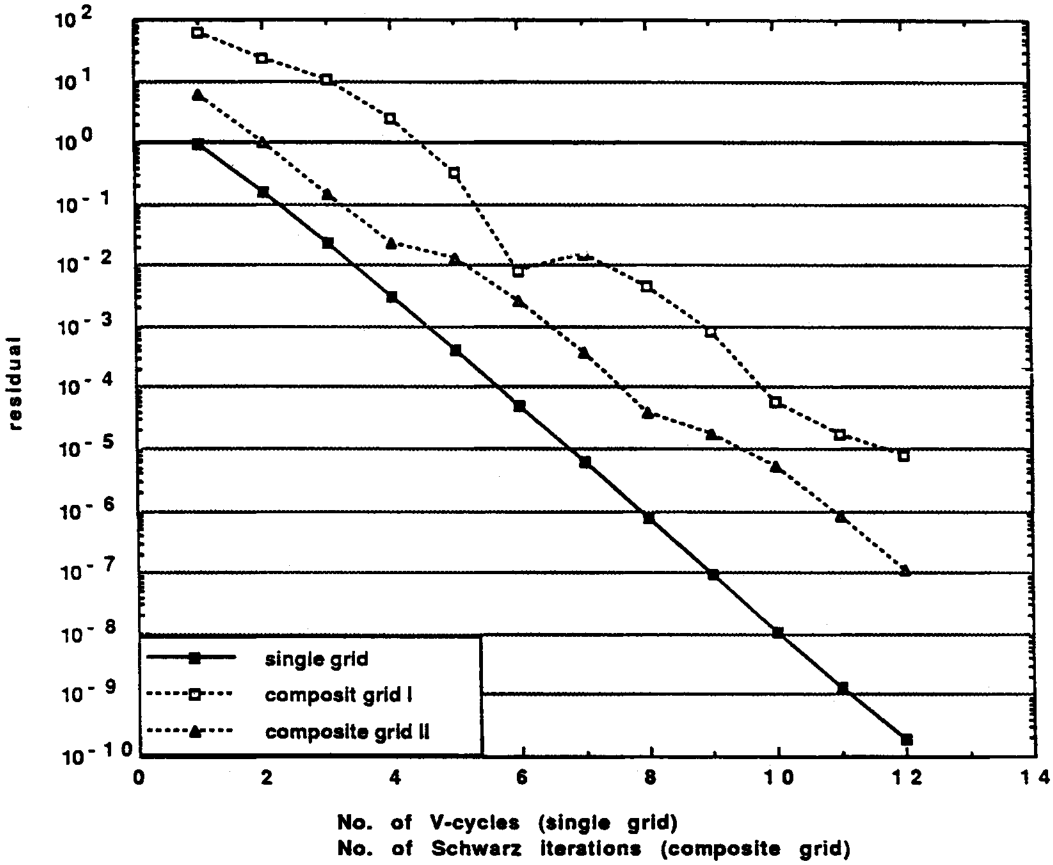


Figure 7. Comparison of convergence rates of the non-composite and composite multigrid methods

conditions on a staggered grid, it is difficult to give boundary conditions for the error. Also because Neumann boundary conditions are applied on the physical boundary, the programming is simpler if Neumann IBCs are used. Finally, as shown in the 1D test problem, Neumann IBCs give faster convergence, so this is the choice we shall make.

Now let us describe the implementation of the interior Neumann boundary conditions. As mentioned in Section 2, we used fictitious points outside the computational domain. In Figure 9, suppose points A and B are members of grid I contained in grid II. In the ICMG method, values at A and B are obtained by interpolating grid II data. The boundary condition for the error requires its gradient to be zero.

In the CCMG method we need coarse grid boundary information as well. To find the boundary condition at point E in Figure 9, the error at points F and G is interpolated. Further, the solution on the fine grid at points A, B, C and D is interpolated to give the derivative at points H and I.

For interpolation we used the Coons patch technique[4] with second-order accuracy. Although analysis indicates that a third-order method should be used, we find that this method is sufficiently accurate; it does, however, lead to some minor discrepancies in the results which will be discussed below. Figure 10 shows the stencil for second-order Coons patch interpolation. P in Figure 10 is the point at which interpolated data are required; α and β are the local co-ordinates

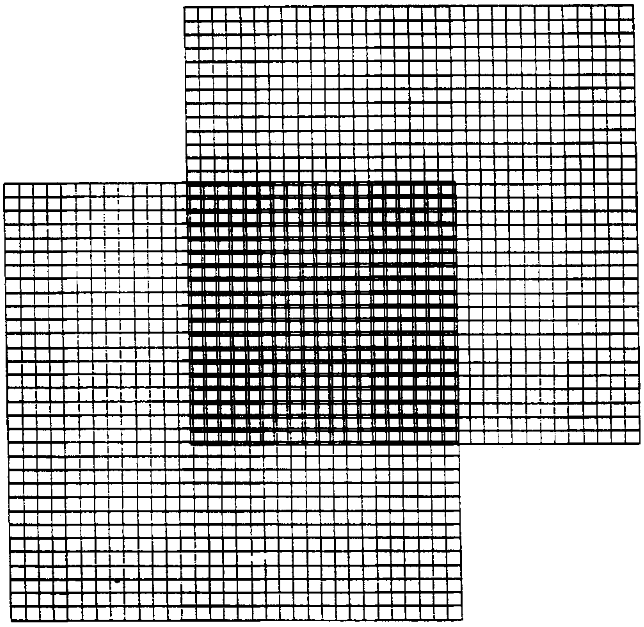


Figure 8. Physical domain and mesh for the 2D composite grid test computation and the box–box cavity flow

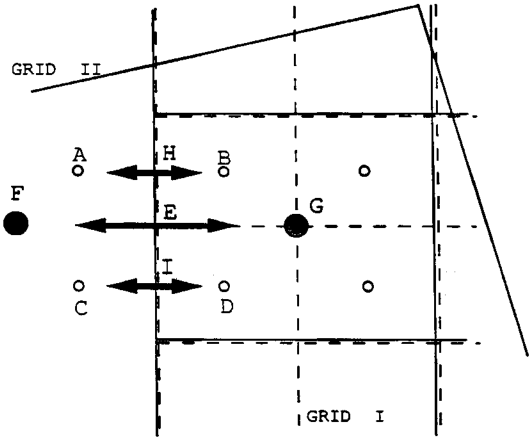


Figure 9. Stencil for interpolation of interior boundary conditions

of P. The interpolated data can be expressed as

$$\phi(P)=(1-\alpha)D+\alpha B+(1-\beta)A+\beta C-(1-\alpha)(1-\beta)\phi(\mathrm{a})-(1-\beta)\alpha\phi(\mathrm{b})-\alpha\beta\phi(\mathrm{c})-(1-\alpha)\beta\phi(\mathrm{d}), \tag{34}$$

where $\phi(\mathrm{a})$ is the value at the corner 'a'. The local co-ordinates (α, β) of the four corners are $\mathrm{a}=(0, 0)$, $\mathrm{b}=(1, 0)$, $\mathrm{c}=(1, 1)$ and $\mathrm{d}=(0, 1)$. A, B, C and D are second-order polynomials along the edges ab, bc, cd and da respectively (see Figure 10).

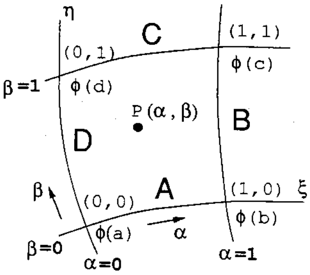


Figure 10. Stencil for Coons patch interpolation

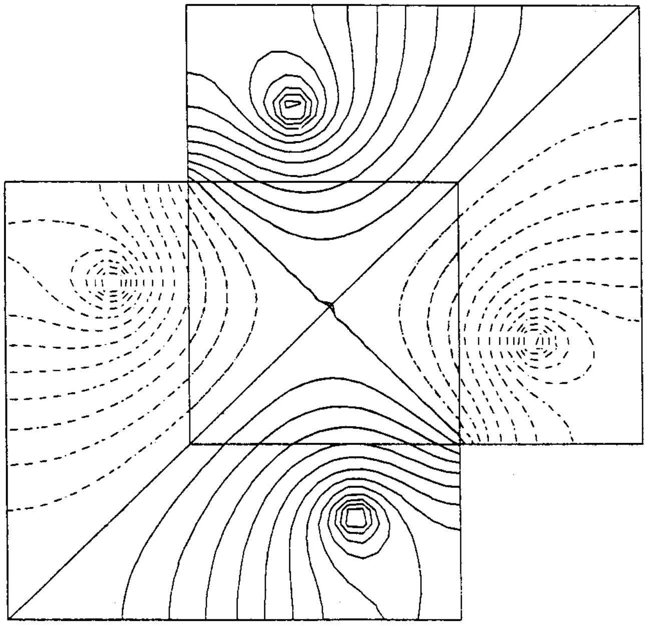


Figure 11. Converged solution to the 2D Poisson equation

We now show the results of a test computation. In Figure 11 the converged solution is shown. The solutions completely coincide in the overlap domain. In Figure 12 the maximum residual in each subgrid is shown as a function of iteration number. The fastest convergence is obtained with the ICMG method with $\omega = 1{\cdot}1$. As in the 1D problem, the effect of ω on convergence is much weaker for the CCMG method than for the ICMG method. Of the interior boundary conditions applied to the CCMG method (30) yields the fastest convergence. The ICMG method converges faster than the CCMG method. Thus we used the ICMG method in the Navier–Stokes solver.

To investigate the speed of the composite grid method relative to a single grid, we solved the four-singularity problem in a square domain. V-cycle iterations were performed for the single-grid method and 10 Schwarz (one V-cycle) iterations were carried out for the ICMG method. Figure 13(a) shows the result obtained using a single (33 × 33) grid and Figure 13(b) shows the

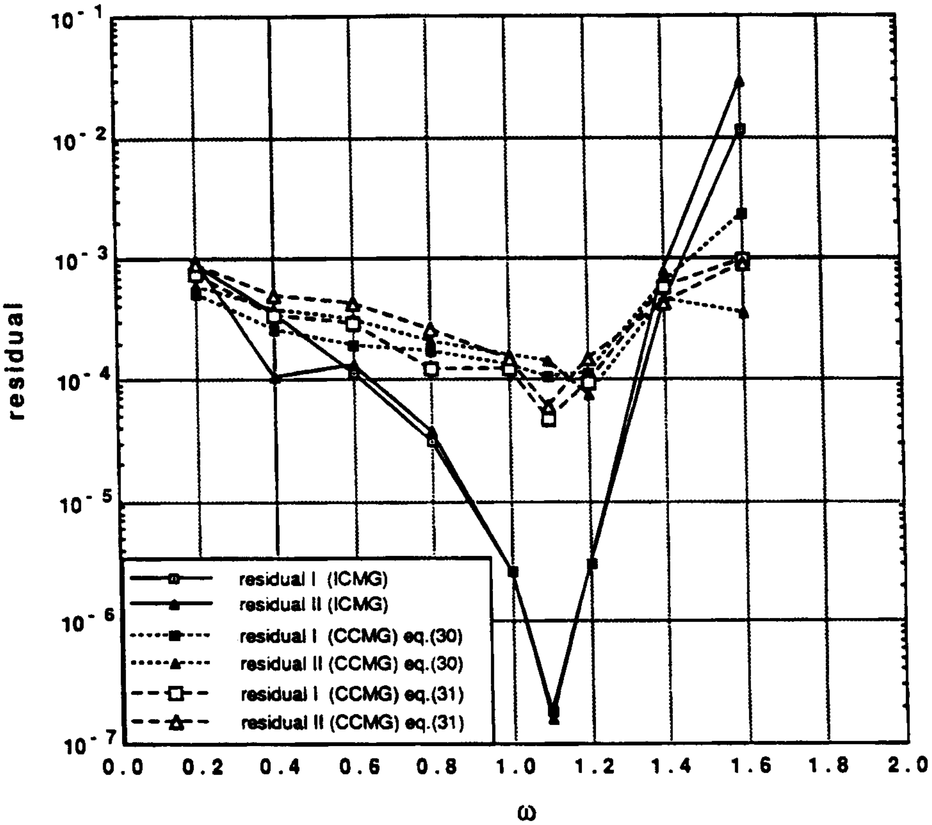


Figure 12. Comparison of the maximum residual after 20 Schwarz one-V-cycle iterations for the ICMG method and two Schwarz 10-V-cycle iterations for the CCMG method in the 2D test problem using Neumann IBCs and Neumann boundary conditions at the domain boundary

result using the ICMG method. The composite grids are each 33 x 25 and, in the overlap region, two subgrids coincide. The two solutions agree. The single-grid method required 29·1 s on a SUN-3 with a floating point accelerator and the ICMG method required 61·5 s. Computations were performed using double precision. The maximum residual after 10 V-cycle computations on the single grid is $4{\cdot}38 \times 10^{-8}$. For the ICMG method the maximum residuals on the two grids after 10 Schwarz iterations are $9{\cdot}43 \times 10^{-6}$ and $4{\cdot}20 \times 10^{-6}$ and the maximum difference of the gradient on the interior boundary is $1{\cdot}17 \times 10^{-8}$.

5. COMPOSITE MULTIGRID METHOD FOR THE N-S EQUATIONS

The composite multigrid method is used to solve the pressure equation. The momentum equations are updated explicitly on the finest grid. Because we use the QUICK scheme, we need two fictitious points outside the boundaries; the velocity components are obtained by interpolation using Coons patches at these points (see Figure 14). There is a difference between the stencils for pressure and velocity interpolation; this is shown in Figure 15. For pressure interpolation the stencil contains nine cells, while for velocity interpolation only one cell is used. Velocity data at the cell corners are found by extrapolating the adjacent velocity data.

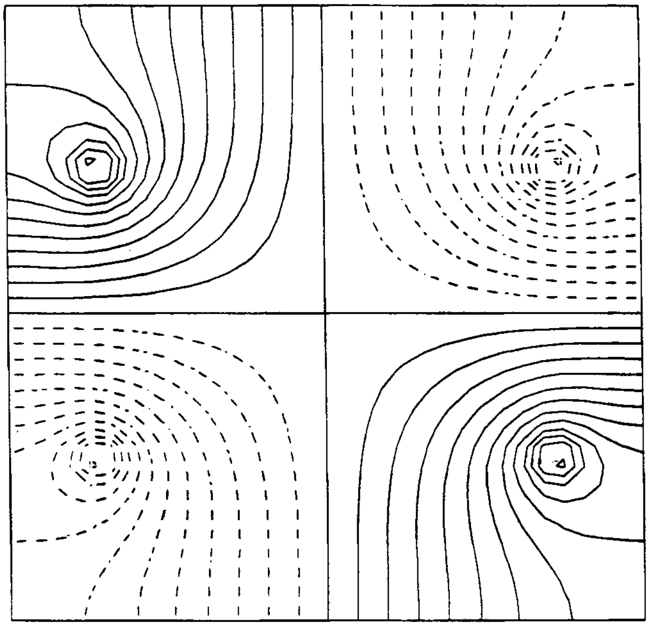


Figure 13(a). Converged solution to the 2D Poisson equation on a single grid

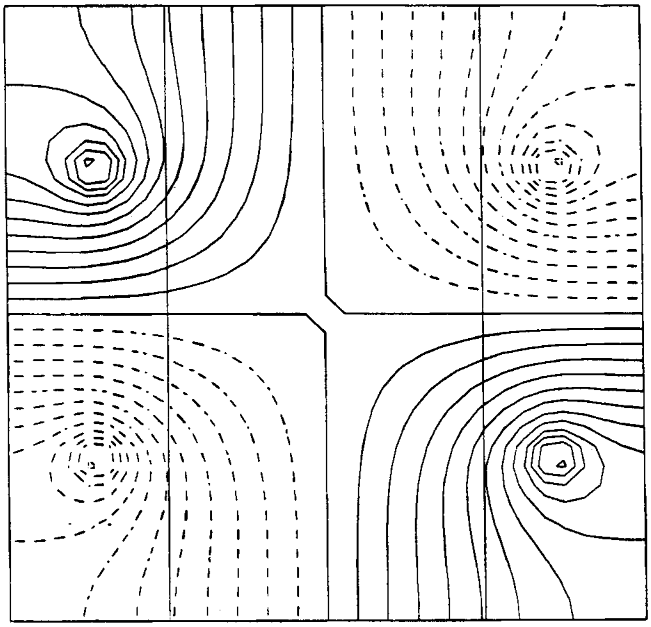


Figure 13(b). Converged ICMG solution to the 2D Poisson equation

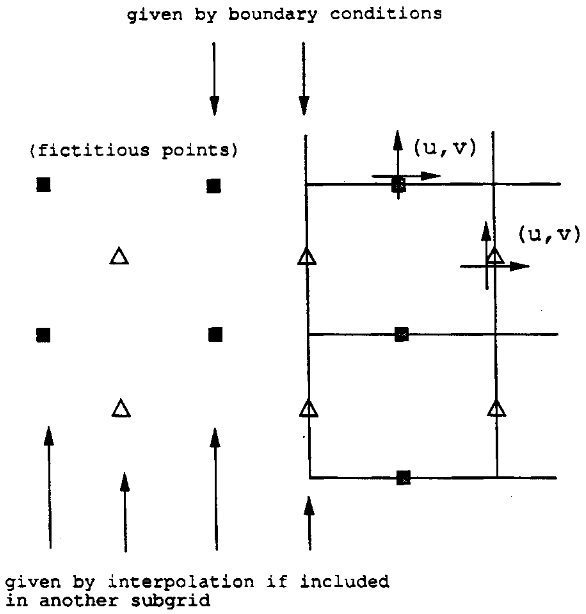

given by boundary conditions
(fictitious points)
(u,v)
(u,v)
given by interpolation if included in another subgrid

Figure 14. Stencils for momentum equation at the boundary

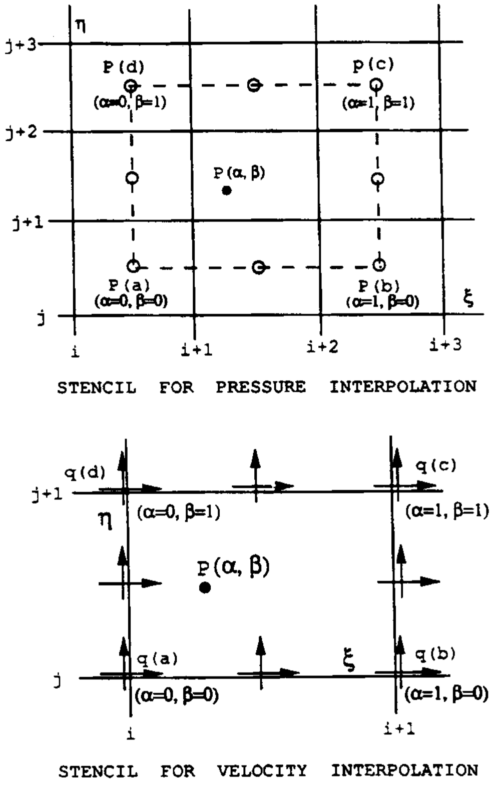


Figure 15. Stencils for Coons patch interpolation of pressure and velocity

6. CHECK OF NUMERICAL SCHEME OF N–S SOLVER

In order to check the computer program, we computed lid-driven cavity flow at $Re=3200$. First we used a single grid. The computations were carried out on 33 x 33, 41 x 41, 49 x 49 and 65 x 65 grids with $\Delta t=0{\cdot}02$, 0·015, 0·015 and 0·01 respectively. The convergence criterion required the maximum increment of velocity in one time step to be less than 1×10^{-6}. A CYDRA-5 minisupercomputer was used; for the 33 x 33 case, 9500 time steps and 160 min of CPU time were required to reach steady state. Except in the early stages of the calculation, in which the pressure solver was limited to five V-cycles, we required that the maximum change in the pressure between iterations be less than 1×10^{-8}.

The convergence error can be estimated as[14]

$$\varepsilon^n \simeq \frac{\phi^{n+1}-\phi^n}{\lambda_1-1}, \tag{35}$$

where λ_1 is the largest eigenvalue (spectral radius) of the iteration matrix. In the Poisson solver with the Gauss–Seidel method it can be approximated by $\lambda_1 \simeq \pi^2/N^2$, N being the number of grid points. For the momentum equations we cannot evaluate λ_1 in advance but, if the maximum

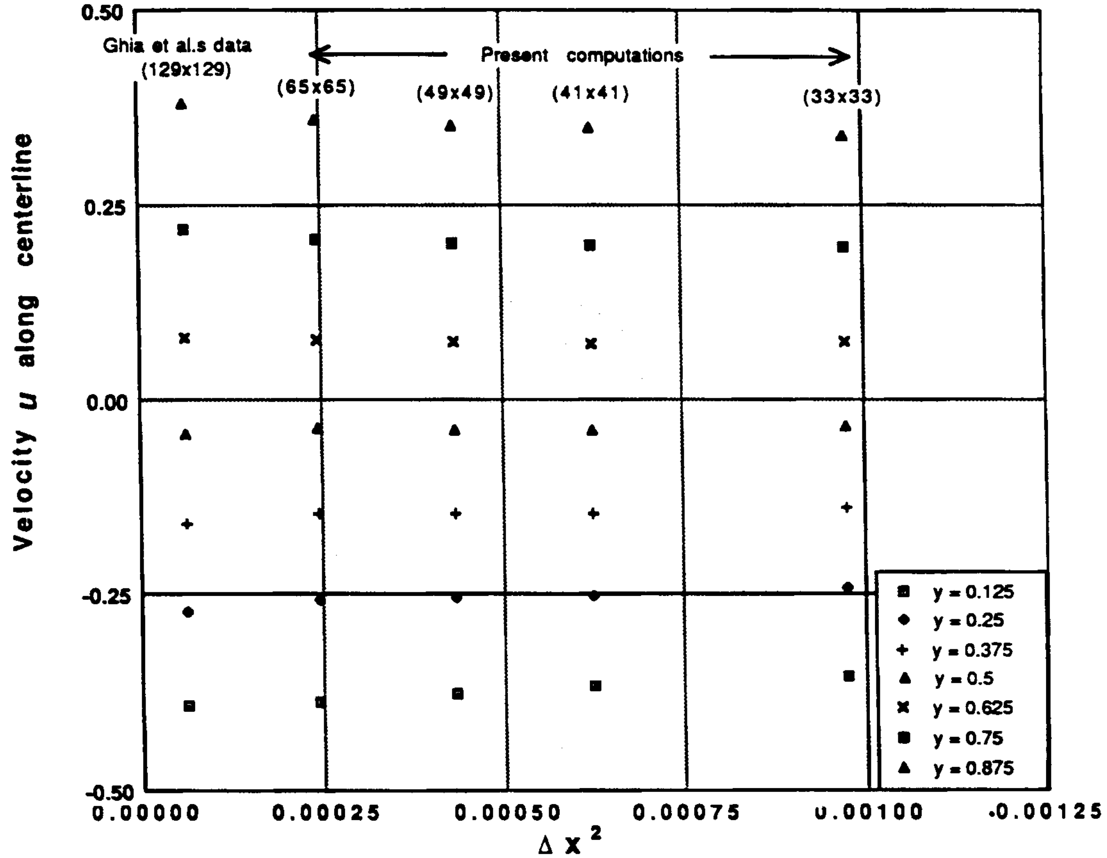


Figure 16. Centreline velocity in the lid-driven cavity flow as a function of grid size

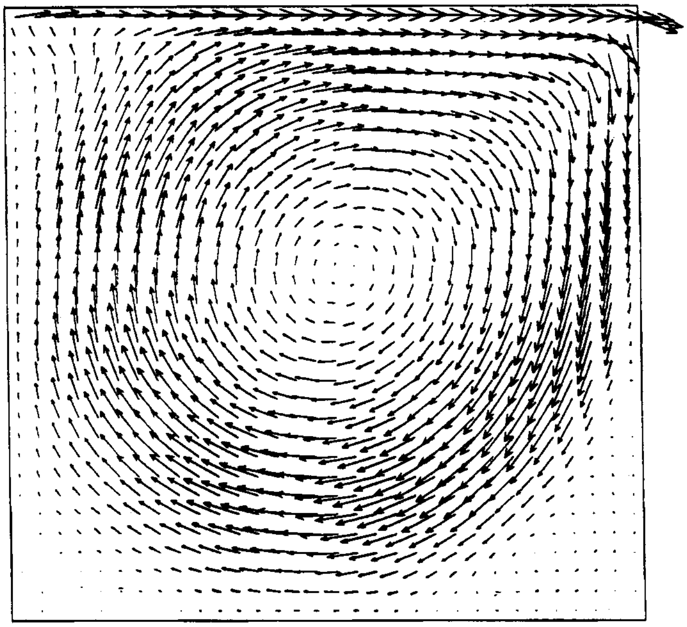


Figure 17(a). Velocity distribution on the centreline of the cavity for the non-composite grid at $Re=1000$

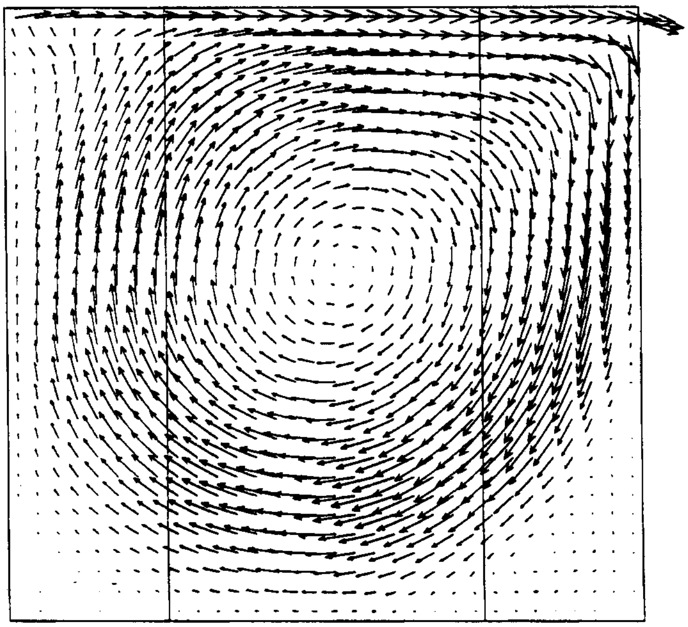


Figure 17(b). Velocity distribution on the centreline of the cavity for the composite grid at $Re=1000$

eigenvalue is real, we can use the estimate[15]

$$\lambda_1 \simeq \frac{\Delta u^{n+1}}{\Delta u^n}. \tag{36}$$

The results show that $\lambda_1 - 1 \simeq O(10^{-3})$. Thus we estimate the convergence error as 10^{-3}.

Since the present scheme has second-order accuracy in space, the solution can be expanded in a Taylor series about the exact solutions:

$$\phi = \phi_{\text{exact}} + a(\Delta x)^2 + b(\Delta x)^3 + \cdots. \tag{37}$$

On a sufficiently fine mesh the error should be proportional to $(\Delta x)^2$. Figure 16 shows the velocity at several points on the vertical centreline as a function of the grid size; the error is proportional to $(\Delta x)^2$ as expected. Results obtained by Ghia *et al.*[16] are also shown. We estimated the exact velocity by Richardson extrapolating the value to $\Delta x = 0$; the values obtained are a little smaller in absolute value than Ghia *et al.*'s values. The difference is largest near the driven lid but is less than 1·5%. The reason for these differences is unknown.

Next we show results of the multigrid composite grid method for cavity flow at $Re = 1000$ on the 33 × 33 grid. The flow domain was decomposed into two subgrids of 25 × 33 grid nodes each; the ICMG method was used to solve the pressure equation with the convergence criterion mentioned above. In this case the two grids coincide in the overlap region; the purpose of this case is merely to demonstrate the composite grid method and the method is essentially the one of Perng and Street.[3] Figure 17(a) shows the velocity profile at steady state on the single grid, while Figure 17(b) shows the velocity distribution found by the composite grid technique; the arrows coincide so completely in the overlap region that they appear to be single arrows. In the single-grid case, 2590 time steps were required to reach steady state with $\Delta t = 0{\cdot}02$; 60 time steps require 1 min of CPU time. For the composite grid case, nine time steps require 1 min of CPU time and 2562 iterations were needed to reach the steady state with $\Delta t = 0{\cdot}02$. The difference is due to the composite grid having 1·5 times as many points as the single grid and, in the single-grid computations, only one V-cycle was allowed per time step, while in the composite grid computations we allowed five Schwarz iterations per time step. These two profiles agree and there is no discrepancy in the overlap region. Thus the accuracy of the scheme is confirmed.

7. APPLICATION TO COMPLEX FLOW FIELDS

Now we show some results for geometrically complex flows. First we solved a lid-driven two-box cavity problem. The physical domain used is the same as in the 2D two-box test computations (see Figure 8). The edge length and lid speed are unity. The upper lid is driven towards the right while the lower lid is driven towards the left. The Reynolds number is 1000. Each square had 33 × 33 nodes. The time step Δt was 0·02, making the Courant number less than 0·64 everywhere. In the computation of the pressure the ICMG method with $\omega = 1{\cdot}1$ was used and five Schwarz iterations with one V-cycle were allowed at each time step. In 1 min, 5·6 time steps are taken on the CYDRA-5.

Figure 18(a) shows the velocity profile at $t = 6$; there are two symmetric vortices. At $t = 9$ the two vortices have grown bigger and the centres closer, as shown in Figure 18(b). Eventually the two vortices merge into one large vortex. The velocity profile at $t = 12$, shown in Figure 18(c), contains one deformed large vortex. This vortex causes two large recirculating flow regions in the corners. Figure 18(d) shows the steady state velocity profile and Figure 18(e) the corresponding vorticity distribution. Figure 18(f) shows the pressure distribution; the pressure contours on the two grids differ slightly. The pressure is obtained by solving (15); the surface integral of the RHS

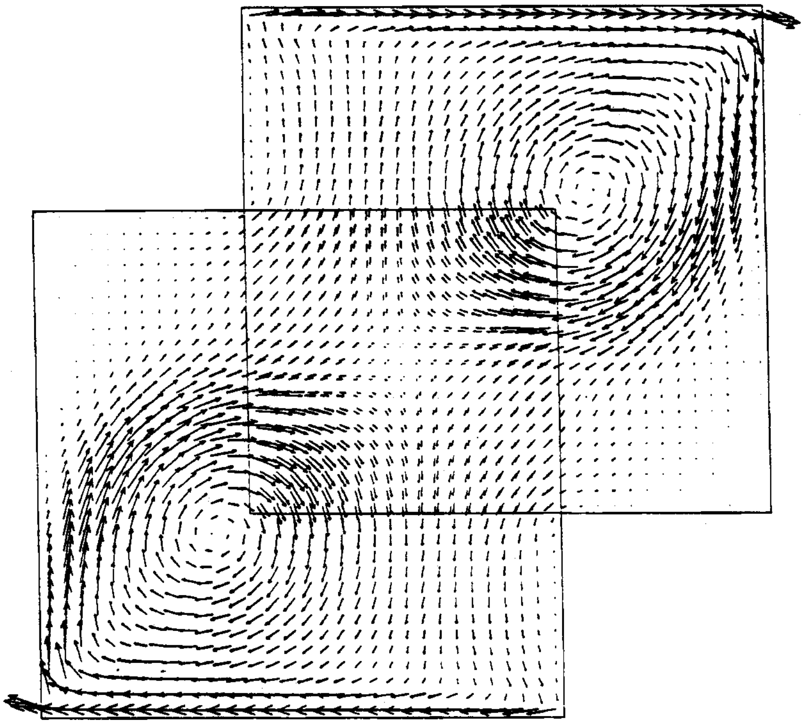


Figure 18(a). Velocity distribution in the two-box cavity flow at $Re=1000$ and $t=6$

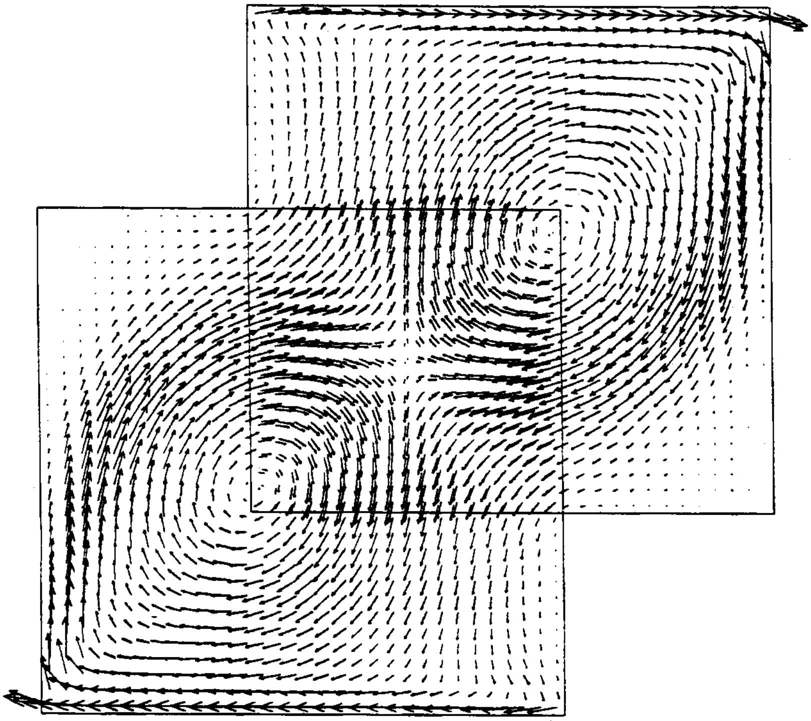


Figure 18(b). Velocity distribution in the two-box cavity flow at $Re=1000$ and $t=9$

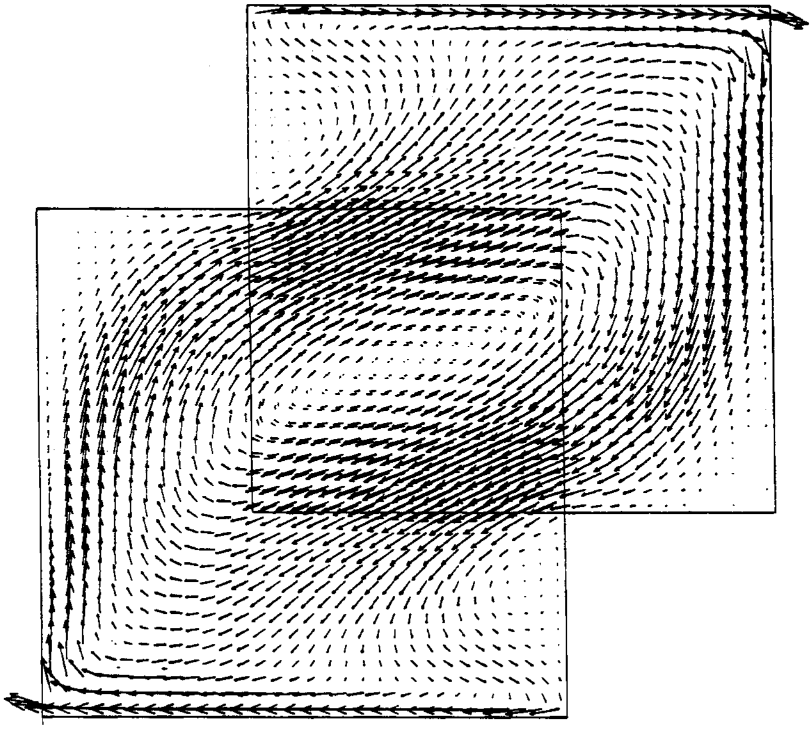


Figure 18(c). Velocity distribution in the two-box cavity flow at $Re=1000$ and $t=12$

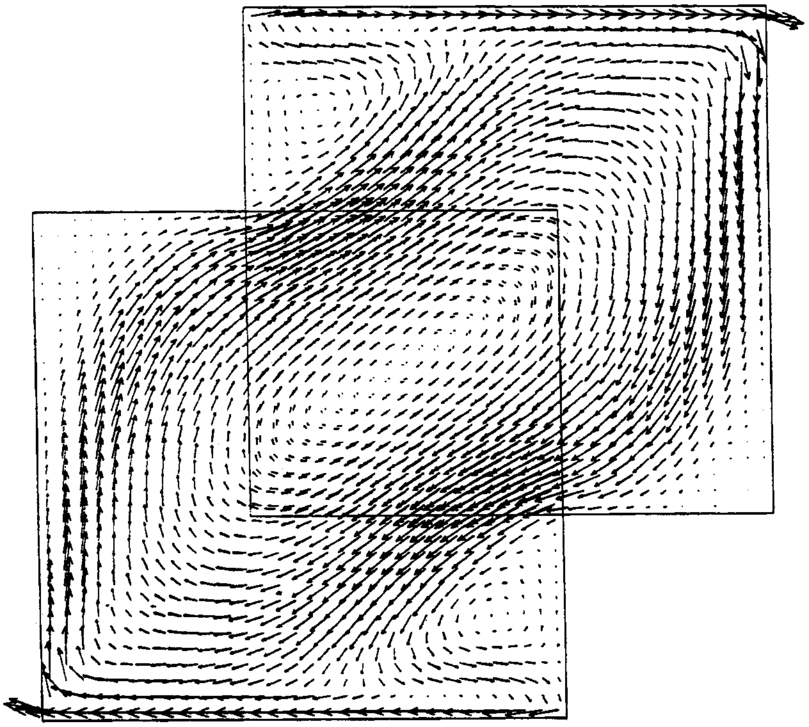


Figure 18(d). Velocity distribution in the two-box cavity flow at $Re=1000$ and $t=60$

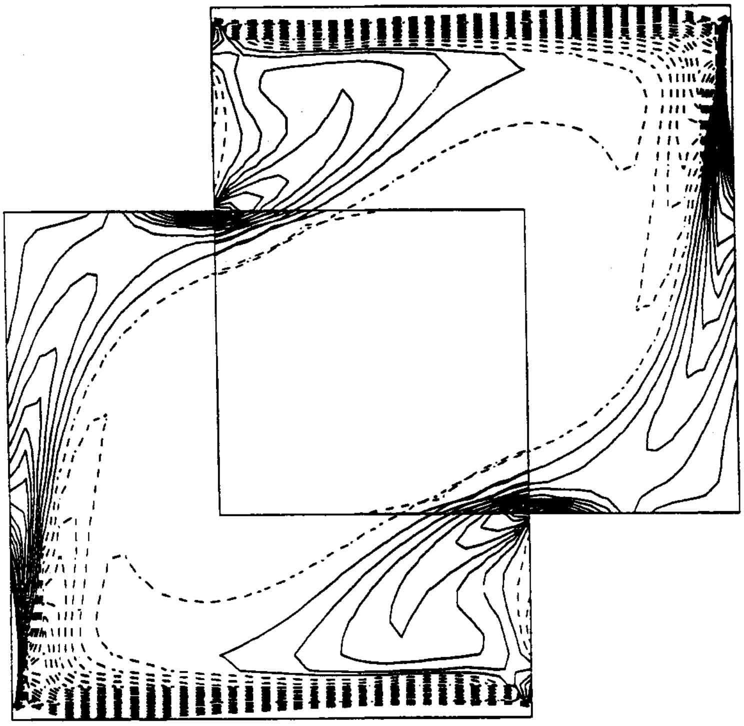


Figure 18(e). Vorticity distribution in the two-box cavity flow at $Re=1000$ and $t=60$; $\Delta\zeta=1$, only contours for $-20\leqslant\zeta\leqslant20$ are shown

of (15) on the overlap domain must be the same on the two subgrids, i.e.

$$\int_{S_{\Omega_{\mathrm{I}}}} \nabla\tilde{U}\,\mathrm{d}S=\int_{S_{\Omega_{\mathrm{II}}}} \nabla\tilde{U}\,\mathrm{d}S. \tag{38}$$

Since the scheme has second-order spatial accuracy, (38) must be obeyed to at least second order and the interpolations on the interior boundaries should be more than second-order-accurate. In the present work we used the Coons patch method with second-order peripheral functions, which is the probable cause of the small pressure discrepancies in the overlap region. In the solution of the Poisson equation, for which the results are shown in Figure 11, the Coons patch appears not to cause any discrepancy in the overlap region. We believe that this is due to the absence of sources in the overlap region. On the other hand, in the Navier–Stokes solution the Poisson equation contains sources in the overlap region and the second-order accuracy of the interpolation scheme produces small discrepancies; however, these are of the order of the accuracy of the scheme and are not important. The errors might become significant if a flow with high-frequency oscillations were simulated with this method.

Next we show results for an annulus–box combined cavity flow. Figure 19 shows the flow domain and grid. The Reynolds number is 1000 based on the length of the square edge. The outer lid of the annular section was driven counterclockwise and the left edge of the square section moved vertically upwards. Figures 20(a) and 20(b) show the velocity and vorticity distributions at $t=49$. Two large stable vortices occupy the flow domain. In the overlap domain the velocities on the two grids agree very well. This shows the usefulness of the composite grid strategy for geometrically complicated flows.

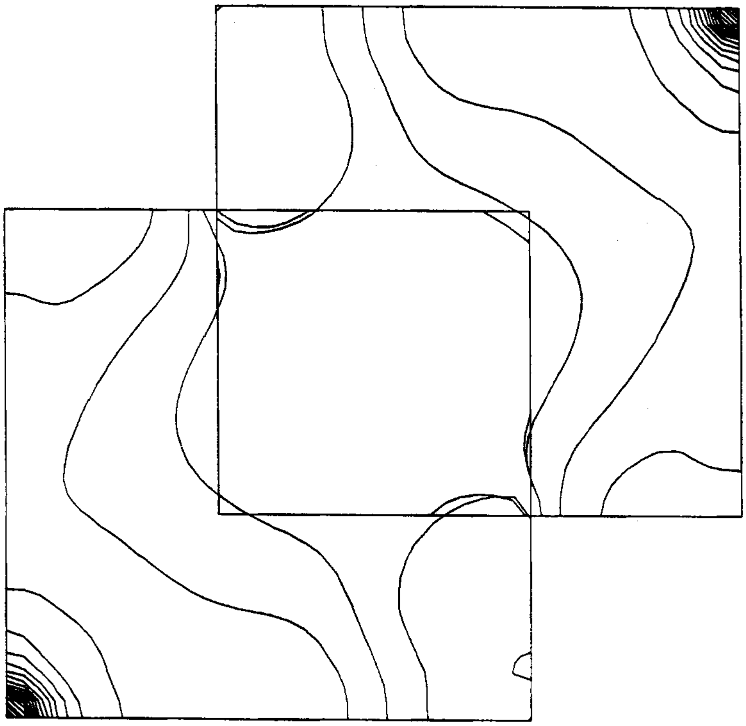


Figure 18(f). Pressure contours in the two-box cavity flow at $Re=1000$ and $t=60$; $\Delta p=0{\cdot}02$

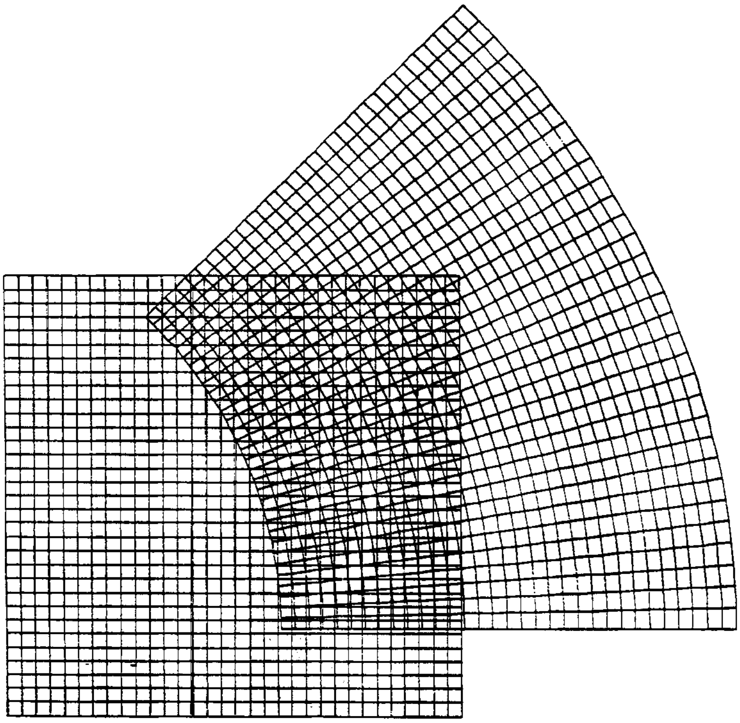


Figure 19. Domain and grid for the annulus–box cavity flow

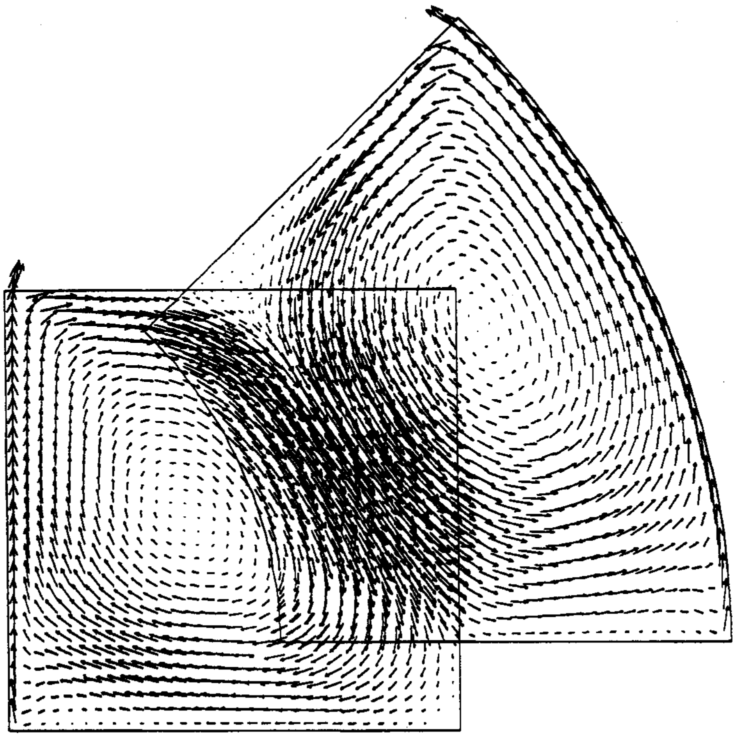


Figure 20(a). Velocity distribution in the annulus–box cavity flow at $Re=1000$ and $t=49$

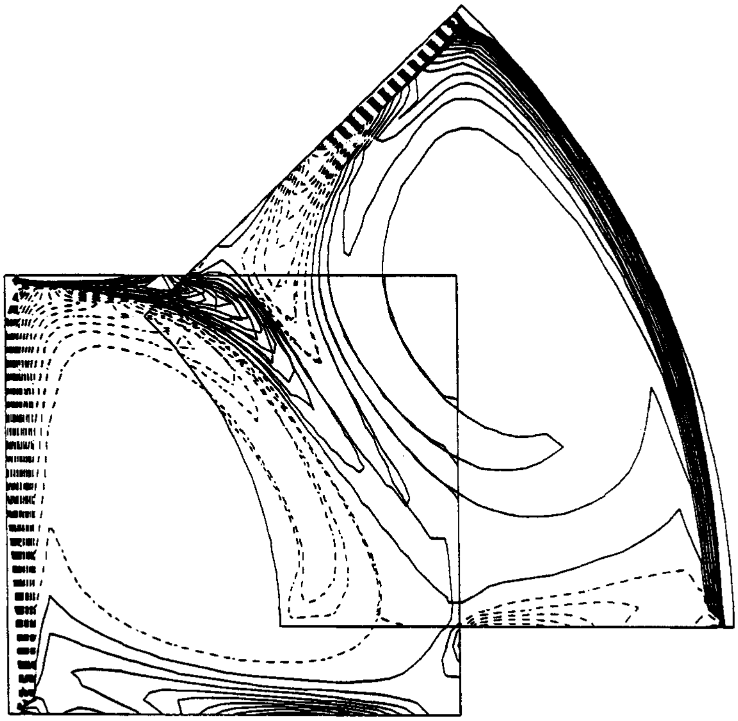


Figure 20(b). Vorticity distribution in the annulus–box cavity flow at $Re=1000$ and $t=49$

8. CONCLUSIONS

To simulate unsteady flows in geometrically complex domains, we discretized the 2D unsteady Navier–Stokes equations using a staggered grid in curvilinear co-ordinates; the accuracy is second-order.

The effectiveness of the composite multigrid approach for geometrically complex flows was demonstrated. It may be used both for flows in which an accurate time history is required or for the computation of steady state flows. We investigated the convergence of the ICMG and CCMG methods for a 1D problem and showed that acceleration of the interior boundary values is very effective. The optimal value of the acceleration parameter depends on the boundary conditions. In the 2D problem the ICMG method again has better convergence than the CCMG method; however, acceleration of the ICMG method is also effective.

Finally we simulated two complex cavity flows at $Re=1000$ to demonstrate the effectiveness of the method for solving the Navier–Stokes equations in geometrically complex domains.

ACKNOWLEDGEMENTS

This work was carried out during the first author's stay at Stanford University under the support of the Science and Technology Agency of Japan. He would like to express his thanks to STA for giving him the chance to visit Stanford. The authors received useful suggestions through many discussions with Dr. Robert van der Wijngaart, Dr. Chin-Yuan Perng, Mr. Ramana Venkata and Professor J. Oliger.

REFERENCES

1. A. Brandt, *Multigrid Techniques: 1984 Guide, with Application to Fluid Dynamics*, von Karman Institute Lecture Series, Brussels, 1984.
2. H. A. Schwarz, 'Über einige Abbildungsaufgaben', *J. Reine Angew. Math.*, **70**, 105–120 (1869).
3. C. Y. Perng and R. L. Street, 'A domain decomposition technique for solving geometrically complex flow problems', *Forum on Numerical Simulation of Convection in Electronic Equipment Cooling, ASME Winter Annual Meeting, Heat Transfer Division, Vol. 121*, December 1989, pp. 61–68.
4. I. D. Faux and M. J. Pratt, *Computational Geometry for Design and Manufacture*, Ellis Horwood, Chichester, 1979, pp. 198–203.
5. C. R. Maliska and G. D. Raithby, 'A method for computing three dimensional flows using non-orthogonal boundary-fitted coordinates', *Int. j. numer. methods fluids*, **4**, 519–537 (1984).
6. B. P. Leonard, 'A stable and accurate convective modelling procedure based on quadratic upstream interpolation', *Comput. Methods Appl. Mech. Eng.*, **19**, 59–98 (1979).
7. R. W. MacCormack, 'The effect of viscosity in hypervelocity impact cratering', *AIAA Paper 69-354*, 1969.
8. R. F. van der Wijngaart, 'Composite-grid techniques and adaptive mesh refinement in computational fluid dynamics', *Ph.D. Thesis*, Stanford University, January 1990.
9. P. L. Lions, 'On the Schwarz alternating method III: a variant for nonoverlapping subdomains', *Proc. 3rd Int. Symp. on Domain Decomposition Methods for Partial Differential Equations*, SIAM, Philadelphia, PA, 1990, pp. 202–213.
10. R. L. Meakin and R. L. Street, 'Simulation of environmental flow problems in geometrically complex domains. Part 2: A domain-splitting method', *Comput. Methods Appl. Mech. Eng.*, **68**, 311–331 (1988).
11. W. D. Henshaw and G. Chesshire, 'Multigrid on composite meshes', *SIAM J. Sci. Stat. Comput.*, **8**, 914–923 (1987).
12. C. Y. Perng, 'Adaptive-multigrid computations for incompressible flows, including geometry, temperature, and salinity effects', *Ph.D. Thesis*, Stanford University, May 1990.
13. W. P. Tang, 'Schwarz splitting and Template operators', *CLaSSiC Report 87-3*, Department of Computer Science, Stanford University, 1987.
14. J. H. Ferziger, 'Estimation and reduction of numerical error', *Forum on Methods of Estimating Uncertainty in Fluid Flow Computations, ASME Winter Annual Meeting*, San Francisco, CA, December 1989, pp. 1–8.
15. G. D. Smith, *Numerical Solution of Partial Differential Equations: Finite Difference Methods*, 3rd edn, Clarendon, Oxford, 1985, pp. 272–273.
16. U. Ghia, K. N. Ghia and C. T. Shin, 'High-*Re* solutions for incompressible flow using the Navier–Stokes equations and a multigrid method', *J. Comput. Phys.*, **48**, 387–411 (1982).