INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 13, 269–286 (1991)

A COUPLED MULTIGRID–DOMAIN-SPLITTING TECHNIQUE FOR SIMULATING INCOMPRESSIBLE FLOWS IN GEOMETRICALLY COMPLEX DOMAINS

C. Y. PERNG* AND R. L. STREET

Environmental Fluid Mechanics Laboratory, Stanford University, Stanford, CA 94305, U.S.A.

SUMMARY

This paper describes a domain decomposition numerical procedure for solving the Navier–Stokes equations in regions with complex geometries. The numerical method includes a modified version of QUICK (quadratic upstream interpolation convective kinematics) for the formulation of convective terms and a central difference scheme for the diffusion terms. A second-order-accurate predictor–corrector scheme is employed for the explicit time stepping. Although the momentum equations are solved independently on each subdomain, the pressure field is computed simultaneously on the entire flow field. A multigrid technique coupled with a Schwarz-like iteration method is devised to solve the pressure equation over the composite domains. The success of this strategy depends crucially on appropriate methods for specifying intergrid pressure boundary conditions on subdomains. A proper method for exchanging information among subdomains during the Schwarz sweep is equally important to the success of the multigrid solution for the overall pressure field. These methods are described and subsequently applied to two forced convection flow problems involving complex geometries to demonstrate the power and versatility of the technique. The resulting pressure and velocity fields exhibit excellent global consistency. The ability to simulate complex flow fields with this method provides a powerful tool for analysis and prediction of mixing and transport phenomenon.

KEY WORDS Unsteady Navier–Stokes Domain decomposition Schwarz QUICK Multigrid

1. INTRODUCTION

Practical and engineering problems in computational fluid dynamics inevitably involve complex geometries. Separation, recirculation and reattachment characterize the general flow structures found on uneven flow boundaries. Sophisticated computational methods are increasingly important for the prediction of flow fields and for the analysis of mixing and heat transfer in regions of flow separation and recirculation. Bhatti and Aung[1] used a hybrid/central differencing scheme to conduct a comprehensive numerical analysis of the two-dimensional forced convection heat transfer in horizontal open cavities. Sparrow and Chuck[2] employed a similar technique to investigate a two-dimensional channel flow with asymmetric abrupt enlargement on an IBM PC. Tamura[3] used a co-ordinate-mapping method to study the aerodynamic behaviour around a square bluff body at high Reynolds numbers. A popular approach employed to treat this type of irregular domain is to use a 'blocking-off' operation[4] to inactivate some of the control volumes of

* Present address: Creare Inc., PO Box 71, Hanover NH 03755, U.S.A.

0271–2091/91/130269–18$09.00

Received 29 August 1989
Revised 27 August 1990

the regular grid so that the remaining active control volumes form the desired irregular domain. This approach, while convenient, is wasteful of CPU time and computer storage. Co-ordinate mapping is sophisticated, but it often becomes cumbersome because extraordinary operations on the governing equations are required. Moreover, some geometries, including the relatively simple ones considered in this paper, are difficult to map into a simple domain. The viability of the aforementioned methods diminishes when the flow domain is long, narrow and/or winding (such as a Z-shaped).

Domain splitting turns out to be simpler to implement, computationally more efficient and more versatile in regions of cut-outs and bends than partial domain inactivation. Furthermore, with the advent of parallel processors in modern computers, techniques of parallelizing computer codes have gained ever-increasing popularity in various computational fields. Domain decomposition techniques hold great promise for being able to take the advantage of parallel computation capabilities so that complex flow problems can be solved far more efficiently. This paper describes a coupled 'multigrid–domain-splitting' technique using an explicit, second-order-accurate predictor–corrector time-marching scheme with modified QUICK/central difference formulation[5] which yields accurate flow fields.

A number of works on the subject of domain decomposition have appeared in the literature. Some deal with linear single-equation models to study characteristics of various methods; some applied the relevant principles to a system of coupled non-linear partial differential governing equations. Oliger *et al.*[6] used a Poisson equation to delve into the Schwarz alternating principle on overlapping grids and explore ways to accelerate its convergence. Henshaw and Chesshire[7] applied a multigrid method to solve a Poisson equation with periodic and Dirichlet boundary conditions on general composite overlapping meshes. Good convergence rates were obtained with the methods described therein. Meakin and Street[8,9] combined boundary-conforming co-ordinate and domain decomposition principles to solve three-dimensional environmental flow problems. Caruso[10] used adaptive grid calculations (which require multiple, intersecting domains) to simulate two-dimensional elliptic flows. Implicit schemes which take each subdomain as an individual unit are used and, subsequently, Schwarz-like iterative methods were employed to sweep through all the subdomains, while a line-by-line method was used for relaxation within each region. Each iterative sweep needs recalculations of the momentum and pressure equations on an individual subdomain.

The strategy employed in this study is in sharp contrast to traditional implicit schemes in many ways. The treatment of the pressure field and its boundary conditions is crucially important to the success of the explicit scheme in solving decomposed flow domains. Instead of being treated as a local property to each individual domain, the pressure field is viewed as a global property of the entire flow field and is treated as a whole in this paper. Once the momentum equations have been calculated on each subregion, the relevant information is incorporated into the source terms of the pressure equation. We then employ a multigrid technique together with a Schwarz-like iterative method to sweep through subregions to solve the pressure equation. The pressure field obtained in this way is globally consistent over the entire flow domain. There are advantages to this technique. First, the momentum equations are calculated only once for each stage of predictor and corrector, unlike implicit methods which need to recalculate these equations for each iterative sweep; therefore our technique involves far fewer algebraic operations within each time step. Secondly, since the pressure field is treated globally, the resulting velocity field retains a global consistency on each subdomain. As a consequence, the velocity profiles in overlapping zones of different subdomains match perfectly. Finally, the global pressure field is very efficiently solved by the multigrid solver. Therefore the entire computational procedure is accurate as well as cost-effective.

Because more general (irregular or curvilinear) domains can be reduced by co-ordinate mapping and domain decomposition to a set of intersecting rectangular domains,[8,9] it is not necessary to deal with irregular or curvilinear domains to demonstrate our method. The illustrative configurations considered in this paper include a square insert located at the lower left corner of a square driven lid cavity, and a two-dimensional channel with abrupt enlargement and contraction at bends. The results are presented using particle track plots which clearly show pronounced flow structures such as primary and secondary recirculating eddies and regions of separation and reattachment in the flow field. This paper proceeds by discussing, first, the numerical methods, secondly, the multigrid technique and its implementation to composite domains and, thirdly, the results of numerical computation. Finally, conclusions are summarized.

2. NUMERICAL METHODS

2.1. Governing equations

The governing equations together with discretizing methods, modified QUICK, solution algorithm and notations for the explicit scheme have been previously described in Reference 5; only relevant procedures are elaborated in this paper. Briefly, in this study we use a volume-averaged formulation to solve a weak form of the two-dimensional Navier–Stokes equations in primitive variables on a staggered grid system in Cartesian (x, y) co-ordinates. The equation for pressure P is obtained by substituting u- and v-velocity components into the discretized continuity equation. This procedure represents a combination of conservation of momentum and mass.

Throughout this study the flow is assumed to be two-dimensional and laminar. The final discretized momentum equations for a uniform mesh take the following forms for the explicit, second-order-accurate predictor–corrector time-stepping scheme:

predictor

$$u_{ij}^{n+1/2} = u_{ij}^{n} + FX_{ij}^{n} + \frac{\Delta t}{2\rho\Delta x}(P_{ij} - P_{i+1,j})^{n+1/2}, \tag{1}$$

$$v_{ij}^{n+1/2} = v_{ij}^{n} + FY_{ij}^{n} + \frac{\Delta t}{2\rho\Delta y}(P_{i,j} - P_{i,j+1})^{n+1/2}; \tag{2}$$

corrector

$$u_{ij}^{n+1} = u_{ij}^{n} + FX_{ij}^{n+1/2} + \frac{\Delta t}{\rho\Delta x}(P_{ij} - P_{i+1,j})^{n+1}, \tag{3}$$

$$v_{ij}^{n+1} = v_{ij}^{n} + FY_{ij}^{n+1/2} + \frac{\Delta t}{\rho\Delta y}(P_{i,j} - P_{i,j+1})^{n+1}; \tag{4}$$

where

$$FX^{n+t_f-1/2} = \{\sum A_{np}u_{np} + b_u^* - u_P\sum A_{np}\}^{n+t_f-1/2} \Big/ \left(\frac{\rho\Delta V_P}{t_f\Delta t}\right), \tag{5}$$

$$FY^{n+t_f-1/2} = \{\sum A_{np}v_{np} + b_v^* - v_P\sum A_{np}\}^{n+t_f-1/2} \Big/ \left(\frac{\rho\Delta V_P}{t_f\Delta t}\right). \tag{6}$$

Here t_f is a time factor; $t_f = 1/2$ for the predictor stage and $t_f = 1$ for the corrector. The A_{np} are the influence coefficients resulting from discretization of the momentum equations. The subscript 'np' denotes a neighbour of a centre node 'P' and the summation is taken over all four neighbours in the 2D case; b_u^* and b_v^* are the appropriate source terms in the discretized momentum equations.

The predictor–corrector scheme employed in this work uses a combination of two explicit time steps. However, for convenience of explanation, a two-dimensional incompressible flow with explicit Euler time marching is used for illustration in this section. Then, equations (1)–(4) are simplified and become

$$u_{ij}^{n+1} = u_{ij}^n + FX_{ij}^n + \frac{\Delta t}{\rho \Delta x}(P_{ij} - P_{i+1,j})^{n+1}, \tag{7}$$

$$v_{ij}^{n+1} = v_{ij}^n + FY_{ij}^n + \frac{\Delta t}{\rho \Delta y}(P_{i,j} - P_{i,j+1})^{n+1}. \tag{8}$$

Continuity requires a divergence-free flow field for time step $(n+1)\Delta t$, i.e.

$$[u_{ij} - u_{i-1,j}]^{n+1}\Delta y - [v_{ij} - v_{i,j-1}]^{n+1}\Delta x = 0. \tag{9}$$

By substituting equations (7) and (8) in equation (9), after some manipulation one obtains the discretized pressure equation

$$\left[\frac{\Delta y}{\Delta x}(P_{i+1,j} + P_{i-1,j}) + \frac{\Delta x}{\Delta y}(P_{i,j+1} + P_{i,j-1}) - 2\left(\frac{\Delta y}{\Delta x} + \frac{\Delta x}{\Delta y}\right)P_{ij}\right]^{n+1}$$

$$= [u_{ij} + FX_{ij} - u_{i-1,j} - FX_{i-1,j}]^n \frac{\rho \Delta y}{\Delta t} + [v_{ij} + FY_{ij} - v_{i,j-1} - FY_{i,j-1}]^n \frac{\rho \Delta x}{\Delta t}. \tag{10}$$

The right-hand side of equation (10) contains the source terms of the pressure equation. They consist of flow variables at time step $n\Delta t$ and are known. By appropriately specifying pressure boundary conditions at time step $(n+1)\Delta t$, one can solve equation (10) by any standard method.

2.2. *Boundary conditions for the pressure equation*

2.2.1. On a physical boundary. Equation (10) is valid for interior as well as for boundary nodal points provided that one can calculate the terms FX and FY on the boundaries. For example, in Figure 1 at node $(2, j)$ on domain 2, equation (10) becomes

$$\left[\frac{\Delta y}{\Delta x}(P_{3,j} + P_{1,j}) + \frac{\Delta x}{\Delta y}(P_{2,j+1} + P_{2,j-1}) - 2\left(\frac{\Delta y}{\Delta x} + \frac{\Delta x}{\Delta y}\right)P_{2,j}\right]^{n+1}$$

$$= [u_{2,j} + FX_{2,j} - u_{1,j} - FX_{1,j}]^n \frac{\rho \Delta y}{\Delta t} + [v_{2,j} + FY_{2,j} - v_{2,j-1} - FY_{2,j-1}]^n \frac{\rho \Delta x}{\Delta t}. \tag{11}$$

Node $(1, j)$ is a boundary point, but all other indices are for interior nodal points. Additional information is generally needed for the determination of pressure boundary conditions. A concise and surprisingly simple approach occurs when the normal velocity components on the boundary are known. In this case, at node $(2, j)$, equation (9) becomes

$$u_{2,j}^{n+1}\Delta y - [v_{2,j} - v_{2,j-1}]^{n+1}\Delta x = \underbrace{u_{1,j}^{n+1}}_{\text{known}}\Delta y. \tag{12}$$

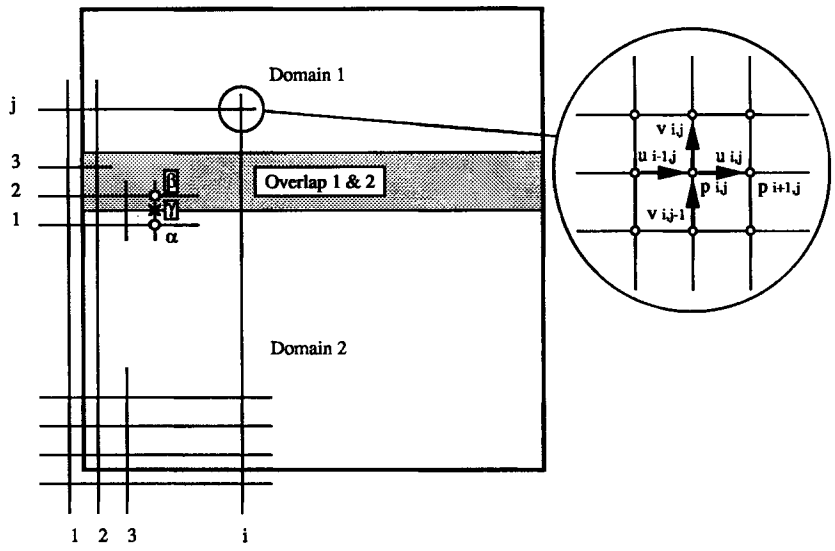

Figure 1. A typical arrangement for two overlapping domains

Consequently, by substituting equations (7) and (8) in equation (12), one obtains the counter part of equation (11):

$$\left[\frac{\Delta y}{\Delta x}P_{3,j}+\frac{\Delta x}{\Delta y}(P_{2,j+1}+P_{2,j-1})-\left(2\frac{\Delta y}{\Delta x}+\frac{\Delta x}{\Delta y}\right)P_{2,j}\right]^{n+1}$$

$$=\frac{\rho\Delta x}{\Delta t}[v_{2,j}+FY_{2,j}-v_{2,j-1}-FY_{2,j-1}]^n+\frac{\rho\Delta y}{\Delta t}[u_{2,j}+FX_{2,j}]^n+\frac{\rho\Delta y}{\Delta t}\underbrace{u_{1,j}^{n+1}}_{\text{known}}. \tag{13}$$

We can see that the boundary pressure $P_{1,j}^{n+1}$ associated with the prescribed $u_{1,j}^{n+1}$ does not appear in the above equation, which means that no information about $P_{1,j}$ is needed in this case. Perhaps it is of interest to examine how this approach works at the corner node, say (2, 2) on domain 2. By the same procedure one obtains

$$\left[\frac{\Delta y}{\Delta x}P_{3,2}+\frac{\Delta x}{\Delta y}P_{2,3}-\left(\frac{\Delta y}{\Delta x}+\frac{\Delta x}{\Delta y}\right)P_{2,2}\right]^{n+1}$$

$$=\frac{\rho}{\Delta t}\{[u_{2,2}+FX_{2,2}]^n\Delta y+[v_{2,2}+FY_{2,2}]^n\Delta x\}+\frac{\rho}{\Delta t}\underbrace{[u_{1,2}\Delta y+v_{2,1}\Delta x]^{n+1}}_{\text{known}}. \tag{14}$$

It is easy to see that we do not need to specify $P_{1,2}^{n+1}$ and $P_{2,1}^{n+1}$ in equation (14). Obviously, whenever the normal velocity components on the boundary are prescribed, the associated pressure nodes just outside the boundary do not appear in the discretized pressure equation (10). Thus, if velocities can be correctly specified, the corresponding pressure differences are not necessary in the pressure equation. However, the net effect of this approach is to require the satisfaction of the well-known compatibility condition for an incompressible flow, which requires the volume integral of the source terr s in the pressure equation to be equal to zero so that a converged solution to that equation can be obtained. These results can also be derived by

application of the original differential equations for momentum and mass conservation in a limit process at a physical boundary, leading to the Neumann boundary condition for the pressure.[11]

2.2.2. On an interior boundary: the impact of domain decomposition. Here we discuss the domain decomposition technique. If we examine equations (7)–(10) closely, we find that all the information needed to solve the momentum equations at time $(n+1)\Delta t$ is readily available from previous calculations at time $n\Delta t$. For this reason the momentum equations can be treated independently on each subdomain. However, when the pressure equation is solved, velocity components normal to flow boundaries at time $(n+1)\Delta t$ are needed. If the flow boundary is a physical one or the normal velocity component at $(n+1)\Delta t$ can be correctly specified, the pressure equation can be solved on each subregion independently; but on non-physical boundaries of subregions inside the flow domain, neither P^{n+1} nor $\mathbf{V}^{n+1}$ can be correctly specified. Traditionally, implicit schemes are used to deal with this problem. Velocities on non-physical boundaries of the subdomain under consideration are taken from appropriate intersecting subdomains. The solution procedures then proceed with some kind of Schwarz sweep* through every subdomain until convergence is reached.[9] Each iterative sweep typically requires solution of the momentum equations and a Poisson-type pressure equation on each region. Because boundary velocities are respecified at each iteration, no pressure boundary condition is necessary for the implicit approach. Here each subdomain is solved independently. However, by using information available in the overlapping zones, we can connect the individual pressure fields into a global one. That is, instead of solving the pressure for each subdomain independently, one treats the pressure as a global property of the entire flow field. The resulting pressure gradients and pressure values are smooth over the computational domain and mass conservation is consistently maintained.

Recall that equation (10) is valid for both interior and boundary nodes. Note also that in Figure 1, because of the staggered grid system, nodes α and β are for scalar quantities, such as pressure and temperature, and node γ is for vector quantities (v-velocity in this example). Node α is both a boundary node $(i, 1)$ to domain 1 and an interior node to domain 2. The vertical boundary velocity of region 1, $v_{i,1}$, sits at nodes γ. We can apply equation (10) to the nodes across the open boundary on domain 1. Two more pieces of information need to be provided in order to solve the pressure equation of domain 1, namely

(i) the value of $(v+FY)^n_{i,1}$, which is not calculated in the discretized momentum equations on the domain, and
(ii) the relation between $P_{i,1}$ and $P_{i,2}$.

Since $(v+FY)^n_{i,j}$ is only computed for the interior nodes on each subdomain, the value of $(v+FY)^n$ on a boundary node, say $(i, 1)$ on domain 1, is available if that node is also an interior one in another subdomain. Therefore an overlapping zone should be provided as a common region for intersecting subdomains, so that whenever information is needed it can be computed from the solution field in the other subregion. In this way, obtaining the above two pieces of information becomes surprisingly simple. For example, at the vector node $(i, 1)$ (i.e. node γ) on domain 1, although $(v+FY)^n_{i,1}$ is not calculated in this domain, it is computed at an interior node in domain 2 and can be directly used in equation (10) on the open boundary of domain 1. Moreover, since the pressure gradient is needed to compute the velocity field, $P_{i,1}$ and $P_{i,2}$ can be

* In the Schwarz alternating method[6] the subdomains are decoupled by using current results in one subdomain to specify necessary boundary conditions on other related subdomains. Then the solution is obtained by solving the problem on each subdomain sequentially, cycling through them until convergence on all domains is obtained.

related by taking the same pressure gradient across nodes α and β on the two domains. The boundary condition on the non-physical boundary thus remains a Neumann type. An alternative to get $P_{i,1}$ is using pressure values directly on the corresponding nodes in another subdomain, i.e. take it directly from node α on domain 2. In this way the pressure boundary condition becomes a Dirichlet type at a non-physical boundary in an intersecting interior domain. As might be expected, it takes fewer iterations for convergence if a Dirichlet rather than a Neumann condition is enforced.

Any classical iterative scheme can be used to solve the pressure equation on each subregion. Since the multigrid method has emerged as one of the most promising iterative techniques for elliptic partial differential equations, with appropriate treatments on intergrid information exchange, we implemented the MG technique in the current code.

Because either pressure gradients or pressure values across the domain boundaries in the overlapping zones are used to transmit information between grids, a global sweep through all the subdomains is required. In this work we perform a Schwarz iterative sweep on one subdomain after another until global convergence is achieved on all subdomains.

2.2.3. Pressure values versus pressure gradients. As mentioned earlier, either pressure gradients or pressure values across the boundary of overlapping subdomains can be used to relate the pressure boundary condition of the two subregions. The question is which choice will be more efficient or more appropriate. If the mesh sizes of the overlapping domains are the same, the pressure as well as velocity nodes on the two subdomains match exactly. It is more convenient and computationally more efficient to use pressure values directly for the boundary node. However, if the mesh sizes of the overlapping subdomains are different, say $\Delta x_2 = 2\Delta x_1$ as shown in Figure 2, it would be advisable to use the same pressure gradients across the boundary. There are two reasons underlying this choice. First, in a staggered grid system the pressure gradient across two adjacent nodes serves as a driving force to induce the flow velocity between them. Since the velocity U is the same for both subdomains, physically we expect the same pressure gradient across the 'fine' and the 'coarse' meshes. Secondly, no interpolation scheme is required and no extraneous error is introduced by enforcing the choice of the same pressure gradient. However, if one wants to use a pressure value instead, an appropriate interpolation scheme must be chosen to accomplish this purpose. Linear interpolation tends to introduce an error larger than the truncation error of the finite difference schemes employed, while higher-order interpolation schemes might produce unphysical wiggles; the solution accuracy is thus compromised. It is a good practice not to use extra interpolation schemes when they can be avoided; using pressure gradients for different mesh sizes serves this purpose.

2.2.4. The size of the overlapping zone and conservation of quantities. The overlapping zone for intersecting subdomains should be only large enough to accommodate all nodal points required

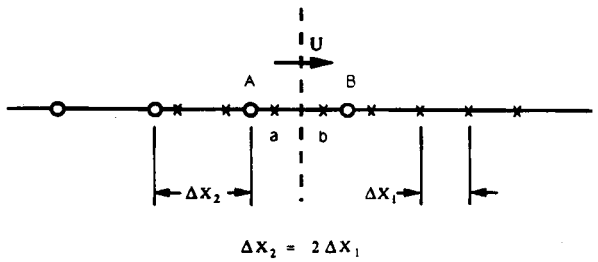

Figure 2. Overlap of coarse and fine grids in a buffer zone

for providing necessary information. Too broad an overlapping zone will not harm the calculation, but it wastes CPU time and other computer resources.

The size of the overlapping zone depends on the choice of schemes one uses for the formulations of spatial derivatives in the Navier–Stokes equations. For example, if a second-order central difference scheme is used to represent the diffusion terms and HYBRID is used for the convective terms, two grid space widths are sufficient for the overlap zone; if one uses QUICK,[12] the zone must be at least three grid spacings wide owing to additional terms which appear in the QUICK formulation. Regardless of the choice of scheme, however, information at a node on a non-physical boundary of one subdomain must be calculated from the solution on an intersecting domain in which that node is an interior point.

As pointed out by Patankar,[4] the most attractive feature of the control volume formulation is that the integral conservation of mass, momentum and energy is exactly satisfied both locally and globally. Also, recall that the pressure equation (10) is obtained by substituting the momentum equations into the continuity equation. Solution guarantees conservation of mass within the accuracy of the solution algorithm.

2.2.5. Algorithm of the technique. We have been using the explicit Euler scheme in previous subsections for the convenience of explanation. In reality we used a predictor–corrector scheme for time stepping. The only change required is to include proper values for the time factor t_f in the equations at each of the predictor and corrector stages. The algorithm for solving the coupled (u, v and p) equations can be summarized as follows.

1. Split the entire irregular domain into several regular subdomains with an appropriate overlapping zone between any two overlapping subdomains.
2. Calculate the momentum equations on each subdomain using equations (5) and (6) with dependent variables at time $n\Delta t$ and set $t_f = 1/2$.
3. Readjust the boundary values of $(u+FX)^n$ and $(v+FY)^n$ on each subregion by using information made available on the appropriate overlapping zones.
4. Apply the Schwarz iteration to cycle through all the subdomains (global iteration) while using SOR, LSOR or MG to sweep over nodal points on each subregion (internal iteration) to calculate the global pressure field (equation (10)) at time $(n+1/2)\Delta t$. The aforementioned pressure boundary conditions should be applied at this step.
5. Update the velocity field at time $(n+1/2)\Delta t$ by substituting the pressure field obtained at step 4 in equations (1) and (2).
6. Recalculate FX and FY on each subdomain using equations (5) and (6) with $t_f = 1$.
7. Readjust the boundary values of $(u+FX)^{n+1/2}$ and $(v+FY)^{n+1/2}$ on each subregion.
8. Repeat step 4 to solve for the global pressure field at time $(n+1)\Delta t$.
9. Update the velocity field at time $(n+1)\Delta t$ using the pressure field obtained at step 8 to conclude one predictor–corrector time step.

There are two advantages to this technique. First, the momentum equations are solved only once on each subregion for each stage of predictor and corrector. Once the information from the momentum equations has been computed, it is then incorporated into the source terms of the pressure equation and we solve for the pressure field. In contrast, in addition to global iterations (Schwarz iteration over every subdomain), implicit methods require both external sweeps (iterations to cycle through momentum, pressure and energy equations) and internal sweeps (SOR over every node within each equation) on every subregion. Each such sweep requires recalculation of both the momentum and pressure equations on each domain. Furthermore, time-accurate solutions require that the local Courant number be of the order of unity; this in turn

limits the time step for a fixed grid system. Therefore our new approach of implementing an explicit scheme with a domain-splitting technique is comparatively cost-effective. (The superiority of using an explicit scheme over an implicit one for the sake of obtaining a time-accurate solution has been reported by Perng and Street.)[5] Secondly, since the pressure field is treated globally, the resulting pressure field varies smoothly over the whole computational domain, i.e. there is no mismatch either in pressure or in pressure gradients at the boundary of two intersecting subregions. As a result, the velocity extracted from this pressure field retains a global consistency and the velocity profiles in the intersecting zones of different subdomains match perfectly.

3. MULTIGRID STRATEGIES

In this section we review the basic concept of the multigrid technique and describe its implementation to solve the pressure equation for incompressible flows in composite domains.

3.1. Basic concept

The multigrid (MG) method has been demonstrated as one of the most efficient iterative techniques for solving elliptic problem. The algorithm of the MG technique is to construct a hierarchy of grids with different mesh sizes. An appropriate relaxation scheme is used as the smoother on each level of grid to quickly reduce large amplitudes of high-frequency error components which cannot be well approximated on the next coarser grid. The 'smoothed' solution is then transferred to the next coarser grid to further eliminate error components with longer wavelengths. By employing several levels of grids, one is able to solve for the high-frequency components on a fine grid and for the low-frequency components on a coarse grid. As a result, the overall convergence rate is greatly accelerated. An excellent introductory discussion of the multigrid method can be found in Reference 13. A more complete introduction to the theories and analyses of the subject is documented in the book edited by Stüben and Trottenberg.[14] References 5 and 15–21 are various applications of the techniques to solve the Navier–Stokes/Euler equations over some simple domains and are good sources to acquire acquaintance with the methods and applications to fluid dynamics.

Since the discretized governing equation for the pressure equation is linear, both CS (defect correction scheme) and FAS (full approximation scheme) are equally applicable. However, the former needs less storage and contains fewer algebraic operation than the latter; therefore we focus our attention on the CS scheme in this paper. The overall procedure of the MG–CS scheme can be described as follows. We first construct a sequence of grids with mesh size $h_k = h_{k-1}/2$ for $k = 2, 3, \ldots, m$, where level 1 is the coarsest grid level and level m is the finest. We then devise two interlevel transfer operators I_k^{k-1} and I_{k-1}^k, where I_k^{k-1}, known as a restriction operator, is used to map variables defined on grid level k onto grid level $k-1$, and I_{k-1}^k, known as an interpolation operator, is used to transfer variables defined on grid level $k-1$ back to grid level k. The solution procedure starts on the finest grid level $k = m$. We first rewrite the pressure equation as

$$L^k P^k = F^k, \tag{15}$$

where L^k is a difference operator containing all the relevant coefficients in the pressure equation, P^k ($k = m$) represents the final converged solution to equation (15) and F^k is the appropriate source term. Then we proceed as follows.

1. Apply an appropriate smoother to equation (15) for a few iterations until the convergence slows down. One gets a defect solution, say p^k, to equation (15) and denotes $V^k = P^k - p^k$ as the defect (error) value.

2. Switch to the next coarser level of grid, $k-1$, and solve there the defect equation

$$L^{k-1}V^{k-1} = I_k^{k-1}(F^k - L^k p^k), \tag{16}$$

where V^{k-1} is the approximate value of the error V^k on the grid level $k-1$.

3. Apply the smoother to equation (16) and obtain an approximate solution v^{k-1} to it. At this stage some criteria are evaluated to determine whether to switch to coarser grids and repeat the above procedure or to interpolate v^{k-1} back to grid k and make corrections on p^k by

$$p^k_{\text{new}} = p^k_{\text{old}} + I^k_{k-1}v^{k-1}, \tag{17}$$

where $I^k_{k-1}v^{k-1}$ is an approximation to the defect V^k on level k.

4. Repeat the above steps until a converged solution on the finest level is obtained.

From the above procedure we can see that we solve for the defect term on a coarse grid and make corrections on a fine grid. This is the basic idea of the MS–CS scheme.

3.2. V-cycle and convergence parameters

Cycle is a term used to denote a specific pattern in which different levels of grids are cycled through and operated on by the chosen smoother. For example , in a V-cycle the process moves from the finest grid to the coarsest and back to the finest again. The cycle pattern can be either explicitly specified by the users or dynamically determined by the programme. By appropriately specifying the convergence parameters δ nd η,[13] the MG algorithm will dynamically determine the patterns of the cycle. δ is the parameter set up to determine the criterion of 'slow convergence rate' on a specific grid level. If the ratio of residual norms at two consecutive iterations is greater than δ, it is considered slow in convergence rate and the smoothing operation on this level should be terminated and switched to the next coarser level. η is used to determine when the solution on the coarsest grid level is converged and the solution procedure should return to the finer grid level. If the residual norm on the coarsest grid level has been reduced by a factor of η, convergence is considered to have been achieved. $\delta = 0{\cdot}6$ and $\eta = 0{\cdot}2$ are commonly used for single-domain computation. Usually, smaller η-values result in a W-cycle. Generally speaking, the W-cycle performs better than the V-cycle on a simple domain. This is not necessarily the case for composite domains. In the current computer code we force the MG algorithm to perform a V-cycle by inserting a few statements in the programme.

There is one additional observation worth mentioning. Instead of using a δ-value (such as 0·6) for the MG algorithm to dynamically switch a smoothing process to the next coarser level (it usually takes two sweeps on the current level before switching) as it does in a single-domain computation, one may force the MG algorithm to perform only *one* iteration on each level except for the coarsest one. This also means that only the η-value is used to determine the convergence on the coarsest grid. At least for the test composite-domain problems (Cases A and B in Section 4), we observed that, by enforcing this strategy, a CPU time saving of 16% was obtained for the two-subdomain case (three levels of grids are used for each domain) and up to 20% was saved for the three-subregion case (four levels of grids are used) over the first 60 time steps from start-up. The saving occurs because performing more than one iteration on each level may or may not reduce the number of Schwarz sweeps; in the meanwhile, extra iterations require more algebraic operations in one V-cycle, leading to greater global computational costs.

3.3. Smoothers and boundary interpolation

The choice of an efficient smoother (relaxation scheme) for a particular problem can significantly influence the convergence rate of the MG iterations. A subdomain grid can be smoothed by

using any standard relaxation scheme. Experience shows that choice of a proper smoother is more or less problem-dependent. Furthermore, some robust smoothers may be able to achieve convergence within a few iterations, but each iteration contains a larger number of operation counts; some smoothers may need more iterative counts, but fewer operations are needed in each iteration. The current code used in this study has been programmed in such a way that a number of available smoothers can be invoked with only minor modifications. The smoothers include Gauss–Seidel, red–black relaxation, line Gauss–Seidel, alternating line Gauss–Seidel, zebra line smoother and alternating zebra line smoothers. One advantage of using the red–black or the zebra scheme is that they are fully vectorizable and parallelizable, thus taking full advantage of computers having vector or parallel processing capabilities.[16]

Three and four levels of grids are used in this study for different geometries. Test runs of current problems on a Cydra-5 mini-supercomputer show that Gauss–Seidel and red–black relaxations are almost equally efficient, while all the smoothers involving line relaxations consume more CPU time to bring the overall pressure field to convergence on the composite domains. It is noted that line relaxations and alternating line relaxations do have better performance when applied to a single (or simple) domain. The degrading of performance for line smoothers over composite grids may be attributed to the fact that the interpolation and exchange of boundary information among component grids cannot catch up with the convergence rate on each grid, which in turns increases the number of sweeps; each sweep of a line smoother contains more operations than point relaxations. Indeed, Stüben and Trottenberg[14] noted that for 'Poisson-like' equations with five-point stencil, red–black relaxation probably yields the most efficient smoother of all.

Interpolating and exchanging boundary information are of primary importance to the efficiency of the composite-multigrid technique. Oliger *et al.*[6] pointed out that the number of iterations required by the 'classic' Schwarz alternating method (which drives the equations to a tight convergence on each subdomain before updating intergrid boundaries) to achieve global convergence depends on the amount of grid overlap and the number of subdomains. Meakin and Street[9] conducted intensive numerical experiments on the subject, solving the Navier–Stokes equations on two overlapping subdomains in a driven lid cavity. They noted that the more frequently the information is exchanged among overlapping domains, the less expensive the corresponding global calculations are. Stüben and Trottenberg, on the basis of numerical experiments on a single Poisson equation with Dirichlet boundary condition, also indicated that the solution efficiency of a 'naive' combination of classic Schwarz iteration with multigrid techniques is limited by the convergence properties of the Schwarz method. They thus proposed a 'direct' smoothing strategy which updates intergrid boundary information right after every smoothing step on each subdomain. Their results showed that the convergence factors are insensitive to the amount of grid overlap. Henshaw and Chesshire[7] followed the same guidelines on their test problems (solving a single Poisson equation over different composite grids) and reached similar conclusions. Because we are solving the Navier–Stokes equations, the information-swapping strategy employed in this study originally stems from Meakin and Street's observations; it turns out to be similar to the idea used in References 7 and 14, but the implementation is different.

Instead of interpolating boundary information right after smoothing a component grid on each grid level, we update intergrid boundaries on the component grid only at the stage after the MG procedure finishes one V-cycle on each subdomain and before it switches to the second finest level. The overall process can be summarized as

(i) perform one MG V-cycle on first subdomain
(ii) update intergrid boundary information
(iii) perform one MG V-cycle on second subdomain
(iv) update intergrid boundary information

and so forth. The major advantage of this approach is that interpolation and exchange of information only occur on the finest level; thus fewer algebraic operations are involved. In addition, one sweep of an MG V-cycle on a component grid can eliminate error components to some degree, and we exchange intergrid boundary information on the finest grid level (where we obtain the final converged solution) right after it finishes one V-cycle. The frequency of information exchange is thus at its highest possible rate. Therefore making use of the information from this partially smoothed grid reduces the total number of Schwarz sweeps and contributes to a better overall convergence rate.

Since the pressure equation for incompressible flows is an elliptic one, the information within the computational domain is transmitted with virtually infinite speed. Therefore, in addition to updating intergrid boundaries as often as possible, the sweeping sequence over subdomains can also influence the computational expenses. If we liken each subdomain to a 'single node' in a simple computational domain, then a one-way sweep sequence (i.e. 123 . . . 123 . . .) is equivalent to a kind of Gauss–Seidel sweep; new information from the previous subdomain is always used by the current subdomain. Since red–black relaxation is more efficient than Gauss–Seidel, we expect that red–black 'sequence' should also be so. Meakin and Street[9] showed that the red–black sweep is slightly more efficient than the one-way sweep but far more efficient than the alternating sweep (i.e. 123 . . . 321 . . .). Of course, there is no difference between the above two sweeping sequences for only two subdomains. In the test problem of a $\square$-shaped channel with three subdomains, we observed that the red–black sequence (i.e. 132132 . . .) is indeed the best. About 8% of CPU time has been saved over the first 800 time steps from start-up.

3.4. Restriction and interpolation operators

The restriction operator is used to transfer variables on a fine grid to the next coarser grid whereas the interpolation operator is used to interpolate variables on a coarse grid to the next finer grid. Because of the staggered grid system used in this study, both operators are derived by using a bilinear interpolation.* Figure 3 shows the coarse and fine grid arrangement for a two-level MG computation. The pressure point sits in the centre of each cell and the fluxes are on the appropriate cell boundaries. Let 'f' and 'c' denote fine and coarse grid points respectively and P be a general variable to be operated on by the restriction/interpolation operator. Then, for the interior nodal points, the restriction operator takes the form

$$P_{ic,jc} = \tfrac{1}{4}[P_{if,jf} + P_{if+1,jf} + P_{if,jf+1} + P_{if+1,jf+1}], \tag{18}$$

where $if = 2ic - 2$ and $jf = 2jc - 2$.

The interpolation operator takes the form

$$P_{if,jf} = \tfrac{1}{16}[9P_{ic,jc} + 3P_{ic+1,jc} + 3P_{ic,jc+1} + P_{ic+1,jc+1}], \tag{19}$$

$$P_{if+1,jf} = \tfrac{1}{16}[3P_{ic,jc} + 9P_{ic+1,jc} + P_{ic,jc+1} + 3P_{ic+1,jc+1}], \tag{20}$$

$$P_{if,jf+1} = \tfrac{1}{16}[3P_{ic,jc} + P_{ic+1,jc} + 9P_{ic,jc+1} + 3P_{ic+1,jc+1}], \tag{21}$$

$$P_{if+1,jf+1} = \tfrac{1}{16}[P_{ic,jc} + 3P_{ic+1,jc} + 3P_{ic,jc+1} + 9P_{ic+1,jc+1}], \tag{22}$$

with $if = 2ic - 1$ and $jf = 2jc - 1$.

Since interpolation of intergrid boundary values takes place only on the finest level, not only should the pressure boundary conditions be properly specified as described in the previous

* A biquadratic interpolation operator has also been tested by the authors. No major improvements on either the solution accuracy or the convergence speed were observed on the staggered grid system.

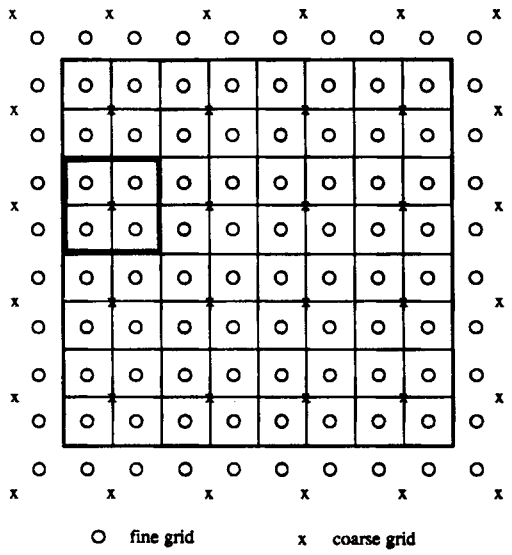

Figure 3. Fine and coarse grid arrangement for MG computation

subsection, but an appropriate restriction operator is also required to transfer the boundary defect to the corresponding coarse grid. To be consistent with the interior scheme, the boundary defect is separately computed and then linearly averaged along the boundary lines to account for the boundary conditions on each level for each component grid. These 'restricted' boundary conditions remain unchanged and will be reinvoked again as the smoothing operator traverses back to this level from the coarsest grid of the same V-cycle.

4. EXAMPLES AND RESULTS

In this section we apply the coupled multigrid, domain-splitting technique to solve two forced convection flow problems with complex geometries. The first one is a square cavity isothermal flow with a square insert at the lower left corner; the second one is an isothermal flow in a two-dimensional channel with abrupt expansion and contraction at the two 90°bends. We regard these geometries as 'complex' because they cannot be easily mapped into a simple domain by standard co-ordinate-mapping methods. The time step Δt for each example is chosen such that the maximum local Courant number is smaller than 0·7 to ensure numerical stability.* The results are presented by using particle track and vorticity plots; typical velocity profiles across different sections of flow domains are also included so that major flow structures can be seen clearly. Note also that in vorticity and pressure contour plots, positive values are drawn with solid lines while negative values are drawn with dotted lines. Comparisons of CPU time required for using SOR on one 'single' level of fine grid (SG) and using the multigrid (MG) technique are listed to show the solution efficiency of the coupled multigrid–domain-splitting technique.

* For the laminar flow calculations reported herein, the Courant limit is considerably more restrictive than the complementary diffusion time step stability limit.

4.1. *Physical examples*

4.1.1. Case A: flow in a driven lid square cavity with a square insert. The dimension of the square insert is one-half of the cavity width so that it blocks out exactly one-quarter of the cavity. The computational domain is divided into two subregions, one containing a 66 × 34 uniform grid and the other 34 × 38. Figures 4(a) and 4(b) show the particle tracks and vorticity respectively of

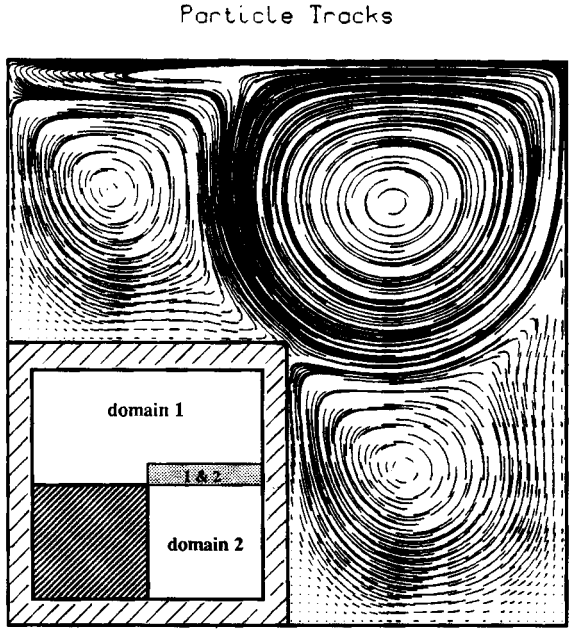

Figure 4(a). Particle tracks for a square insert in a cavity, (Case A), $Re = 3200$

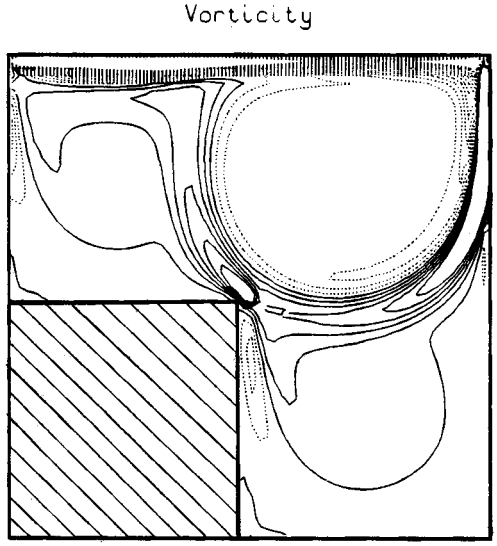

Figure 4(b). Vorticity plot for a square insert in a cavity (Case A), $Re = 3200$

the resulting flow field for $Re = 3200$ (based on the lid velocity and cavity width). From the figures we can see that the flow field exhibits global consistency over the entire domain; fluid particles travel smoothly in and out of the two split domains. Pronounced flow structures can also be observed. First, there are three primary recirculating cells occupying the three major portions of the cavity; three secondary eddies are formed at the three lower corners. Flow separations occur between each pair of primary recirculating cells and between primary and secondary eddies. Secondly, the primary recirculating cell at the upper right region stretches all the way across the width of the cavity beneath the driven lid and forms a kind of bicellular structure. This eddy plays an important role in the momentum transfer from the moving lid to the entire cavity; the other primary cells and secondary eddies are mainly induced by this one.

4.1.2. Case B: a two-dimensional channel flow. The channel has two bends to form a Z-like shape. The channel undergoes an abrupt expansion and contraction in its width right at each bend. An expansion and contraction ratio of two is used for the configuration. Since the steady state flow field is of interest in this type of flow, fully developed boundary conditions are prescribed at the inlet and outlet of the channel. Since the velocities are known at the boundaries, the appropriate pressure equations are modified as in equation (13). The length between the inlet and the expansion section is taken to be twice the channel width so that the flow condition around the first bend will not influence the specified inlet velocity, while the length from the contraction section to the outlet is taken to be 10 times the channel width so that the flow can regain its fully developed stage at the outlet. The height of the expanded region is 5·625 times the channel width to provide enough room for observing flow structures developed in this section. The computational domain is decomposed into three subregions. The first ranges from the entrance to the downstream wall of the expanded region and the second covers the expanded region, ranging from the first bend to the second bend. The third domain is a counterpart of the first one; it contains the rest of the channel. The same mesh size is used in the three subdomains. The numbers of grids in the three subregions are 66×18, 34×66 and 194×18 respectively.

Figure 5(a) shows the particle tracks of the resulting flow field. Since the main stream of flow undergoes dramatic changes in direction when passing through the expansion and contraction sections, interesting flow structures are developed in this region. As the main stream passes the expanded region, it induces a primary recirculating cell that occupies almost one-half of the expanded region, which in turn induces a weak secondary eddy at the lower left corner. Another secondary eddy is induced by the main stream at the upper right corner. The forming of the primary eddy has the effect of pushing against the streamline of the main stream flow near the contraction corner and narrowing its effective passage. As a result, the main stream of flow gets accelerated around the contraction corner and subsequently induces a comparatively large separation bubble along the upper wall of the contracted section. Figure 5(b) displays the normalized streamwise velocities at several sections along the channel. From the two figures we can see that the flow preserves the specified parabolic velocity profile for some distance from the inlet, then begins to be disturbed by the effect of downstream expansion and bend and gradually changes its original shape. As the flow enters the contraction zone, it undergoes a redevelopment process and finally regains its perfect parabolic profile before reaching the exit of the channel.

4.2. CPU time consideration

Table I compares the computational costs for Cases A and B using single-grid and multigrid computations. The listed CPU time has been normalized by the number of time steps and the total grid points in the overall domain. In Case A three levels of grids are employed for the MG computation, while in Case B four levels of grids are used. For single-grid computations, point

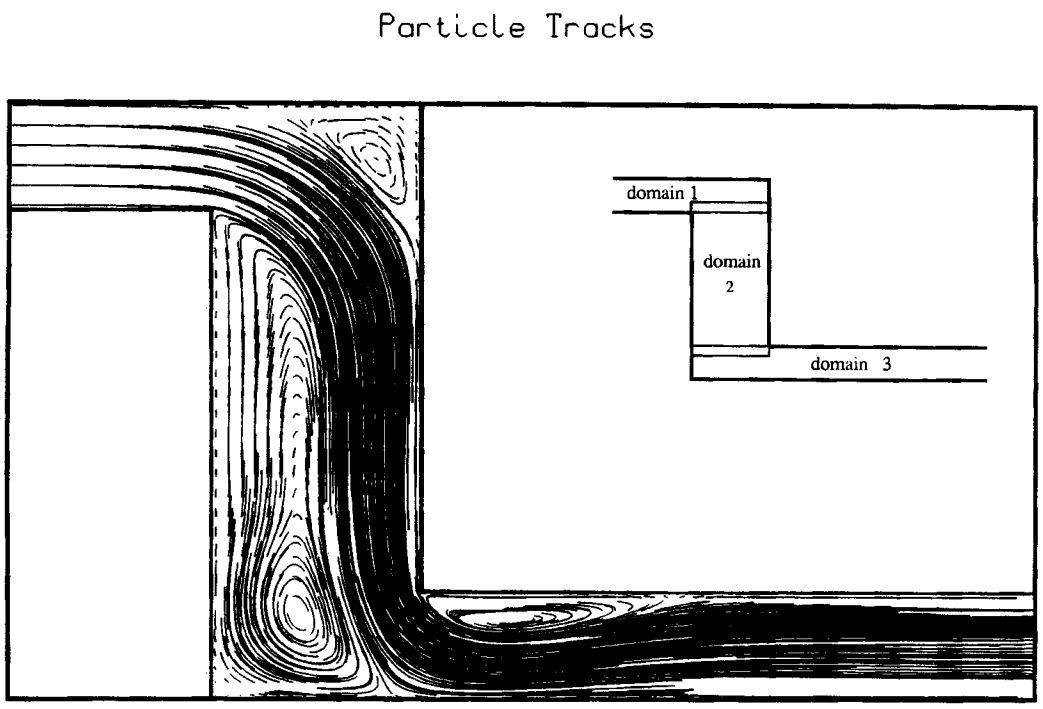

Figure 5(a). Particle tracks showing flow field in a bending channel (Case B), $Re=200$

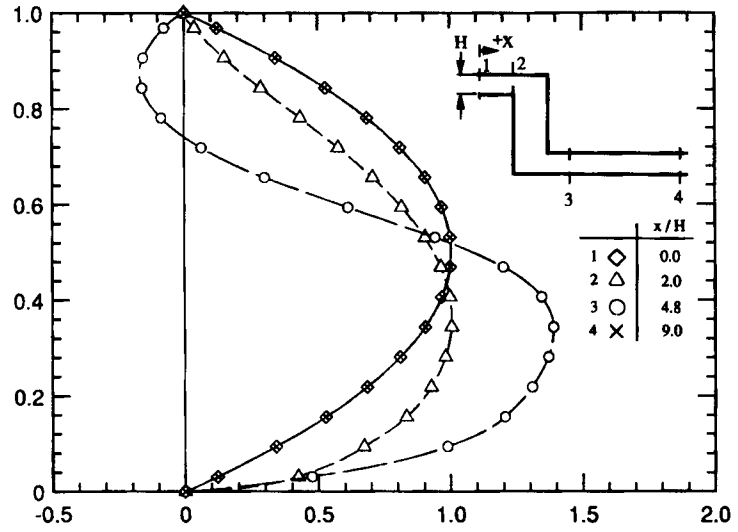

Figure 5(b). Normalized velocity profiles at different sections in a bending channel (Case B), $Re=200$

SOR iteration with an overrelaxation factor of 1·9 (the optimal one in the test runs) is used. All the calculations were carried out on a Cydra-5 mini-supercomputer at Stanford University. The comparisons are made based on the CPU time used over the first 60 time steps, which are equivalent to 120 calls to the pressure-solving routine owing to the predictor–corrector scheme

Table I. Comparisons of computational costs. SG, single grid; MG, multigrid

Problem	Level of grids	Total node points	CPU (SG)	CPU (MG)	Fraction of CPU MG/SG
Case A	3	3536	$2{\cdot}49 \times 10^{-3}$	$7{\cdot}10 \times 10^{-4}$	0·29
Case B	4	6924	$1{\cdot}45 \times 10^{-2}$	$3{\cdot}15 \times 10^{-3}$	0·22

employed in the solvers. The convergence criteria are that the overall residual mass be driven down to 10^{-7} of the initial mass and that relative changes of pressure values in the overlapping zones be less than 10^{-3}. We can see that a significant amount of CPU time has been saved by applying the MG method for the solution of the pressure field over the composite domains.

5. CONCLUSIONS

A coupled multigrid–domain-splitting method for simulating geometrically complex flows is described. The technique is validated by solving a model problem and subsequently applied to three examples of forced convection isothermal flows. A multigrid technique has been implemented to solve the global pressure field over the composite domains. The solution efficiencies of coupling an MG method with composite domains are demonstrated. The velocity and pressure fields obtained exhibit a global consistency over the entire flow domain. We have worked only with regions that can be decomposed into rectangular subdomains. Application of co-ordinate mapping can be used to reduce irregular subdomains to rectangles. Thus any flow in a complex irregular domain can be straightforwardly reduced to the type of cases studied here. The ability to build up complex flow fields in this simple manner provides a powerful tool for analysis and prediction of mixing and transport phenomena.

ACKNOWLEDGEMENTS

The support of the National Science Foundation under Grant No. NSF MSM-8719509 is gratefully acknowledged. The comments of Dr. Stephen Klotz on a draft of this paper are appreciated.

REFERENCES

1. A. Bhatti and W. Aung, 'Finite difference analysis of laminar separated forced convection in cavities', *J. Heat Transfer*, **106**, 49–54 (1984).
2. E. M. Sparrow and W. Chuck, 'PC solution for heat transfer and fluid flow downstream of an abrupt, asymmetric enlargement in a channel', *Numer. Heat transfer*, **12**, 19–40 (1987).
3. T. Tamura, 'Numerical study of aerodynamic behavior of a square cylinder', *J. Wind Eng.*, **37**, 261–270 (1988).
4. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, New York, 1980.
5. C. Y. Perng and R. L. Street, '3D unsteady flow simulation: alternative strategies for a volume-averaged calculation', *Int. j. numer. methods fluids*, **9**, 341–362 (1989).
6. J. Oliger, W. Skamarock and W. P. Tang, 'Schwarz alternating method and its SOR accelerations', *Technical Report CLaSSic-86-12*, Computer Science Department, Stanford University, 1986.
7. W. D. Henshaw and G. Chesshire, 'Multigrid on composite meshes', *SIAM J. Sci. Stat. Comput.*, **8**, 914–923 (1987).
8. R. L. Meakin and R. L. Street, 'Simulation of environmental flow problems in geometrically complex domains, Part 1: A general coordinate transformation', *Comput. Methods Appl. Mech. Eng.*, **68**, 151–175 (1988).
9. 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).
10. S. C. Caruso, 'Adaptive grid techniques for fluid flow problems', *Ph.D Dissertation*, Department of Mechanical Engineering, Stanford University, 1985.

11. P. M. Gresho and R.L. Sani, 'On pressure boundary conditions for the incompressible Navier–Stokes equations', *Int. j. numer. methods fluids*, **7**, 1111–1145 (1987).
12. B. P. Leonard, 'A stable and accurate convective modeling procedure based on quadratic upstream interpolation', *Comput. Methods Appl. Mech. Eng.*, **19**, 59–98 (1979).
13. A. Brandt, 'Multi-level adaptive solutions to boundary-value problems', *Math. Comput.* **31**, 333–390 (1977).
14. K. Stüben and U. Trottenberg (eds), 'Multigrid methods, fundamental algorithms, model problem analyses and applications', *Multigrid Methods, Lecture Notes in Mathematics, Vol. 960*, Springer, Berlin, 1982.
15. A. Brandt, J. E. Dendy Jr. and H. Ruppel, 'The multigrid method for semi-implicit hydrodynamics codes', *J. Comput. Phys.*, **34**, 348–370 (1980).
16. A Brandt, 'Guide to multigrid development', *Multigrid Methods, Lecture Notes in Mathematics, Vol. 960*, Springer, Berlin, 1982, pp. 220–312.
17. 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).
18. C. Y. Perng and R. L. Street, 'The SEAFLOS numerical codes: high speed N–S solvers', *Proc. Third Int. Symp. on Refined Flow Modelings and Turbulence Measurements*, Edited by Y. Iwasa *et al.*, IAHR, Tokyo, July 1988, pp. 191–198.
19. R. E. Phillips and F. W. Schmidt, 'A multilevel–multigrid technique for recirculating flows', *Numer. Heat Transfer*, **8**, 573–594 (1985).
20. V. M. Theosossiou and A. C. M. Sousa, 'An efficient algorithm for solving the incompressible fluid flow equations', *Int. j. numer. methods fluids*, **6**, 557–572 (1986).
21. S. P. Vanka, 'Block-implicit multigrid solution of Navier–Stokes equations in primitive variables', *J. Comput. Phys.*, **65**, 138–158 (1986).