INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 18, 761–780 (1994)

AN ALGEBRAIC MULTIGRID SOLVER FOR NAVIER–STOKES PROBLEMS

R. WEBSTER

Roadside, Harpsdale, Halkirk, Caithness KW12 6UL, U.K.

SUMMARY

An efficient numerical method is presented for solving the equations of motion for viscous fluids. The equations are discretized on the basis of unstructured finite element meshes and then solved by direct iteration. Advective fluxes are temporarily fixed at each iteration to provide a linearized set of coupled equations which are then also solved by iteration using a fully implicit algebraic multigrid (AMG) scheme. A rapid convergence to machine accuracy is achieved that is almost mesh-independent. The scaling of computing time with mesh size is therefore close to the optimum.

KEY WORDS Navier–Stokes Algebraic multigrid Coupled solution Unstructured meshes

1. INTRODUCTION

Spatial resolution is of paramount importance in the numerical solution of fluid transport equations. This is because flow fields of practical interest are usually complex, exhibiting spatial variations over wide ranges of length scale. Even stable laminar flows can require resolution over several orders of length scale.

Resolution is therefore even more important than formal accuracy of a discretization scheme in the sense that high accuracy is of little use if the computational mesh is too coarse to resolve the flow. Since the mesh size requirement (numbers of elements/nodes) scales as the cube of the range of length scales, the need for large computational meshes is inescapable, as is the need for solvers whose efficiency scales well with problem size. Iterative solution methods offer the best prospect in this regard since they offer good scaling of both computing time and data storage with mesh size. Multigrid methods for example can offer close-to-optimum linear scaling.

The computing power required for resolving even moderately complex fluid flows (say two-order resolution in 3D space and in time) is beyond the practical capabilities of serial computing technology and of necessity will require parallel processing. A practical flow solver must therefore be suitable for implementation on parallel processing architectures. Parallel algorithms have already been implemented both for geometric multigrid[1] and for the algebraic multigrid of particular interest here.[2]

Algebraic multigid (AMG) solvers are particularly attractive for the following reasons.

1. 'Coarse grid' matrices are derived automatically by algebraic methods.
2. The flow field itself is accounted for in the coarsening process: coarsening is not simply 'geometrical'.
3. The errors at all length scales are reduced efficiently in the iterative process.

CCC 0271–2091/94/080761–20

Received May 1993
Revised December 1993

This last point is particularly relevant for fine meshes and non-uniform meshes where there is a wide bandwidth of errors. The more complex the flow field, the wider is the range of length scales to be resolved and the wider is the bandwidth of errors to be corrected in the iterative process.

The first attempt to produce an algebraic multigrid solver for fluid flow appears to have been that of Lonsdale.[3] Impressive convergence and close-to-optimum scaling were achieved. However, the method did not prove to be robust.

In the work to be reported here, a different discretization is used and a simpler form of algebraic multigrid for coupled variables is employed. A robust AMG solver for the Navier–Stokes equations is achieved.

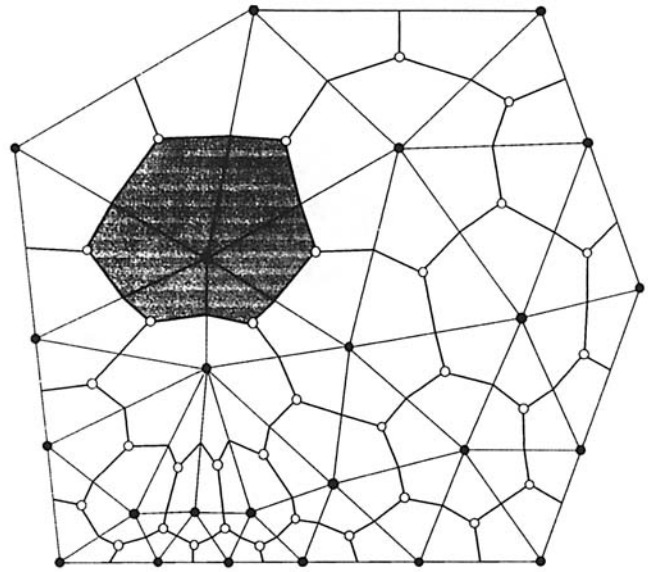
2. DISCRETIZATION

Since the aim is to achieve resolution by mesh refinement, both local and global, discretizations of at least first-order accuracy are essential. Simple linear finite elements will satisfy this requirement, offering between first- and second-order accuracy for Navier-Stokes problems. The simplest possible element is the triangle in two dimensions or the tetrahedron in three dimensions. These also have the added attraction that automatic mesh generation packages have been developed for them.

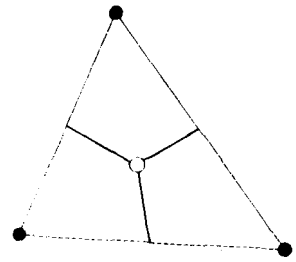
To ensure convergence to accurate solutions, it is important that the discretization produces good coupling between flow and pressure variables. Schemes based on collocated variables may not give this. Odd–even decoupling in the pressure field (checker-boarding) and possible wiggles in the velocity field can occur unless special measures are introduced to prevent them. One such is the addition of a fourth-order dissipation term to the continuity equation,[4] equivalent to a special interpolation for velocity as devised by Rhie and Chow.[5] The penalty is that continuity will not be exactly satisfied owing to the false dilation introduced. For some problems the error can be unacceptably large in regions of high pressure gradient.

In this investigation we look for a compact discretization which is free from such errors but which nevertheless gives good coupling between the variables. This demands different interpolations for pressure p and velocity $\mathbf{u}$. Baliga and Patankar[6] have described two schemes. The first used a coarser discretization for p than for $\mathbf{u}$, the $\mathbf{u}$-elements being nested triangular subelements of the larger triangular p-element. The second due to Prakash[7] and Hookey[8] used equal-order elements with a linear interpolation for p but with a special interpolation for $\mathbf{u}$ based on a local solution of the momentum equation within the element. The latter approach is particularly attractive because the interpolation has a physical basis. Schneider and Raw[9] devised a somewhat similar scheme for rectangular elements and found it gave exceptionally accurate results on standard test problems.

The scheme adopted for this work is essentially that of Prakash and Hookey, though somewhat simplified. The discrete nodal flow variables are collocated at the vertices of the triangular elements. A complete set of discrete equations is assembled for those variables by enforcing the appropriate conservation laws on control volumes constructed for each node (Figure 1). The bounding control surfaces are conveniently formed by the lines joining the element centre to the centres of the element faces (for 3D by also joining the centres of faces to the centres of edges). Values for control surface fluxes can then be calculated from the interpolated flow velocites using the special interpolation. For simplicity a single-element velocity is used, the centre of the element providing the single interpolation point. A convenient solution for this velocity $\mathbf{v}_e$ is a discrete solution in the finite volume approximation. The element itself may be used as the control volume or a subcontrol volume may be constructed. Figure 2 shows one



Control volume tesselation



Linear element

Figure 1. Illustrating the linear triangular element, element assembly and construction of the 'control volume tesselation'; one control volume is highlighted; ○, location of element interpolation point for element velocity $\mathbf{v}_e$; ●, location of nodal velocities $\mathbf{v}$ and pressure $\mathbf{p}$

control volume used for this work, obtained by joining the midpoints of lines joining the interpolation point to the vertices of the element. Other control volumes can be used.

If $\mathbf{u}$ represents the fluid continuum velocity, ρ the density and μ the viscosity, then integrating the momentum flux $\mathbf{f}$ over the surfaces of the control volume and equating it to the integrated volumetric source,

$$\int \mathbf{f} \cdot \mathrm{d}\mathbf{A} = \int [(\rho \mathbf{u} - \mu \nabla)\mathbf{u} + p] \cdot \mathrm{d}\mathbf{A} = \int s \, \mathrm{d}V, \tag{1}$$

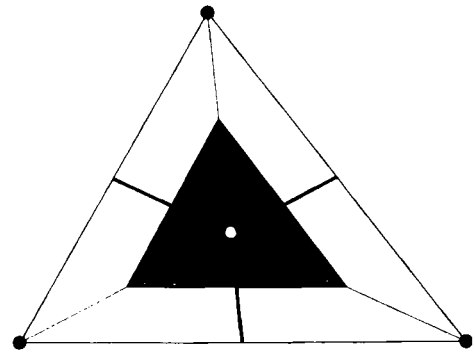


Figure 2. Interpolation for element velocities $\mathbf{v}_e$; a subcontrol volume is used for a local discrete solution of the equation of motion

delivers a global set of discrete equations relating the element velocity array $\mathbf{v}_e$ to the nodal velocity and pressure arrays $\mathbf{v}$ and $\mathbf{p}$ respectively, i.e.

$$A_e(\mathbf{v}_e)\mathbf{v}_e + F(\mathbf{v}_e)\mathbf{v} + G_e\mathbf{p} = \mathbf{s}_e^{\mathrm{m}}, \tag{2}$$

where A_e and F are part of the discrete advection–diffusion operator for elements, G_e is the element gradient operator and $\mathbf{s}_e^{\mathrm{m}}$ is the source array for elements. Enforcing the integral equation (1) for nodal control volumes delivers a second set

$$A(\mathbf{v}_e)\mathbf{v} + G\mathbf{p} = \mathbf{s}^{\mathrm{m}}, \tag{3}$$

where A and G in this case are the nodal advection–diffusion and gradient operators respectively and $\mathbf{s}^{\mathrm{m}}$ is the momentum source/sink array for nodal control volumes.

Enforcing continuity for nodal control volumes provides the required third set

$$D\mathbf{v}_e = 0, \tag{4}$$

where D is the discrete divergence operator.

3. METHOD OF SOLUTION

3.1. Navier–Stokes solution algorithm

The complete set of equations is solved by direct iteration, the advective fluxes being temporarily fixed at each iteration to give a linearized system. The matrix A_e has a simple diagonal form and is easily inverted, so the solution of equation (2) is trivial, i.e.

$$\mathbf{v}_e = A_e^{-1}(\mathbf{s}_e^{\mathrm{m}} - F\mathbf{v} - G_e\mathbf{p}). \tag{5}$$

Substituting directly into the continuity equation (4) delivers a second equation for the nodal variables $\mathbf{v}$ and $\mathbf{p}$ which together with the linearized equation (3) forms the coupled subset

$$\begin{bmatrix} A(\mathbf{v}_e) & G \\ (DA_e^{-1}F) & (DA_e^{-1}G_e) \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{s}^{\mathrm{m}} \\ (DA_e^{-1}\mathbf{s}_e^{\mathrm{m}}) \end{bmatrix}. \tag{6}$$

The temptation to pursue this rationalization further by direct substitution for $\mathbf{v}_e$ in A is

resisted. Since we have to deal with the non-linearity by iteration in any event, the interpolation itself is used once at each iteration to update the flux. The solution algorithm is thus as follows.

1. ASSEMBLE equation (6)
   (using initial guess or latest estimate of $\mathbf{v}_e$).
2. SOLVE equation (6) for $\mathbf{v}$ and $\mathbf{p}$
   (using linear solver).
3. ASSEMBLE equation (2)
   (using latest $\mathbf{v}$ and $\mathbf{p}$).
4. SOLVE equation (2) for $\mathbf{v}_e$
   (i.e. equation 5).
5. IF converged
   STOP
   ELSE
   RETURN to 1.

The explicit interpolation for $\mathbf{v}_e$, step 4, also permits an optional additional step which consists of adjusting the interpolated values of $\mathbf{v}_e$ so that they satisfy continuity to a much tighter convergence criterion. We therefore look for velocity corrections $\mathbf{v}'_e$ *such that*

$$\nabla \cdot (\mathbf{v}_e + \mathbf{v}'_e) = 0 \tag{7}$$

and

$$\nabla \wedge (\mathbf{v}_e + \mathbf{v}'_e) = \nabla \wedge \mathbf{v}_e. \tag{8}$$

The additional constraint ensures that the vorticity will not be disturbed in the adjustment. Equation (8) will be satisfied if

$$\mathbf{v}'_e = -\nabla \Psi, \tag{9}$$

where Ψ is a scalar potential function. Equations (9) and (7) can then be solved for $\mathbf{v}'_e$, substitution of (9) giving a Poisson equation for Ψ. In practice it is convenient to replace (9) by the discrete equation

$$\mathbf{v}'_e = -A_e^{-1} G_e \mathbf{p}', \tag{10}$$

since the discrete equation for the scalar potential $\mathbf{p}'$ has a matrix operator that is then identical to the lower block diagonal in equation (6) and can be assembled from an existing subroutine.

3.2. Linear solver

For the solution of equation (6) a fully implicit solver based on algebraic multigrid methods has been developed. The pioneering work of Ruge and Stuben[10] and later developments of their approach by MacGregor[11] and Lonsdale[3] have provided the basis. As previously noted, Lonsdale produced an AMG solver for Navier–Stokes problems that did not prove to be robust. Complications and approximations arising from a Rhie–Chow-type interpolation are thought to have been responsible. Equation (6) does not contain such complications. Moreover, the new solver does not explicitly use the SIMPLE algorithm as a smoother. Its use is therefore not restricted to Navier–Stokes problems and it promises to be a more general AMG solver for coupled variable sets.

The starting point is the basic scheme for single variables. Though any one of the above schemes could have been chosen. Lonsdale's scheme was adopted because of the appealing simplicity of its coarsening algorithm and the shorter set-up time penalty. It will be helpful to review this basic scheme.

3.2.1. Review of single-variable AMG. Consider the solution of the equation set

$$A\mathbf{x} = \mathbf{b} \tag{11}$$

for the single-variable set $\mathbf{x}$. If $\mathbf{x}^n$ is the latest iterate, we look for a convergent solution process such that the next iterate $\mathbf{x}^{n+1}$ is closer to $\mathbf{x}$. In general $\mathbf{x}^n$ will have a spectrum of errors across a band of wavelengths. For the classical field problems of interest here the bandwidth will depend on the ratio of the largest characteristic dimension in the problem to the closest nodal spacing. Iterative solvers based on point Gauss–Seidel relaxation have an efficiency for reducing errors that depends on the wavelength λ. Efficiency is largest at the shortest wavelength but falls off as λ^{-2}. Thus for wide bandwidth problems (fine meshes) their overall performance is poor.

Multigrid methods have been developed to improve the efficiency bandwidth of the solver. This is achieved by generating a hierarchy of reduced equation sets (coarser grids) such that the shortest wavelengths on each grid progressively span the bandwidth. Application of a point Gauss–Seidel solver to each reduced equation set will thus efficiently remove errors in a succession of wavebands that together span the entire bandwidth.

To generate a coarse grid reduced equation set, a coarsening operator K is applied to give

$$A^c\mathbf{x}^c = \mathbf{b}^c, \tag{12}$$

where

$$A^c = KAK^T. \tag{13}$$

If $\mathbf{b}^c$ is derived on the basis of the residual in equation (11), i.e.

$$\mathbf{b}^c = K(\mathbf{b} - A\mathbf{x}^n), \tag{14}$$

then solution of (12) provides a correction $\mathbf{x}^c$ that can be used to improve the solution

$$\mathbf{x}^{n+1} = \mathbf{x}^n + K^T\mathbf{x}^c. \tag{15}$$

Clearly this procedure can be recursively applied to equation (12) and successively coarser grids (smaller equation sets). Choice of K, choice of the solver on each grid and choice of the strategy for passing residuals and corrections through the grids distinguish the different multigrid methods. For geometric multigrid K is determined entirely from geometric considerations of the computational mesh structure. For algebraic multigrid on the other hand it is determined entirely from considerations of the matrix A. It therefore depends on the physics of the problem and may not have any geometric significance at all; in fact $\mathbf{x}$ need not be a field variable, in which case the coarse meshes may be purely abstract concepts.

Unstructured non-uniform computational meshes pose no particular difficulty for AMG; no special measures are necessary; the algorithm deals with them naturally. Similarly no special consideration needs to be given to boundary conditions; as long as they are implicitly contained in the equation set, the required information will be automatically transferred to the reduced equation sets/coarser grids by the K-operator.

Coarsening algorithm

Essentially the strategy consists of seeking out the nodes with the strongest coupling (largest off-diagonals in A) and joining them together by adding their respective equations. A new coarse grid node is created that is representative of a number of fine grid nodes. The number is arbitrary, but overall a complete coarse grid has about half the number of nodes of the next fine grid. Note that there is therefore no need for the matrix operators K to be formed explicitly. However, if required, they could easily be assembled. Starting from a unit matrix on the finest mesh, they are generated by recursively applying the same A-coarsening operations.

Some care is required in the implementation to avoid long strings of fine grid nodes being collected into a single coarse grid node. Lonsdale avoids this by coarsening in two passes through the nodes, starting from the most strongly connected. On the first pass fine grid nodes are only permitted to join if they have not already been incorporated into a coarse grid node; on the second pass a remaining fine grid node can be attached to the coarse grid node which already contains the neighbour to which the fine grid node is most strongly connected. This second pass is stopped once the sum of the coarse node total and the remaining fine node total reaches the desired size for the coarse grid; unallocated fine grid nodes are then assigned to be coarse grid nodes in their own right. Attention is then turned to the next coarse grid.

The generation of coarse grids stops when no further coarsening can take place. This would be either a single coarse grid node or a number of isolated coarse grid nodes. These final equations would deal with the lowest wave number errors.

Multigrid cycling strategy

The strategy for passing residuals down and corrections up through the grids is the so-called F-cycle strategy summarized in Figure 3. Parallel algorithms for multiple-instruction, multiple-data (MIMD) computing architecture are described in Robinson's paper.[2] Residuals are transferred directly through the grids according to the coarsening rules. Corrections, however, are scaled/damped by a factor β before transfer. For the single-variable solver the correction vector $\mathbf{x}^c$ is scaled such that any remaining error is orthogonal to $\mathbf{x}^c$, i.e.

$$(\mathbf{x}^c, [A(\beta\mathbf{x}^c) - \mathbf{b}^c]) = 0. \tag{16}$$

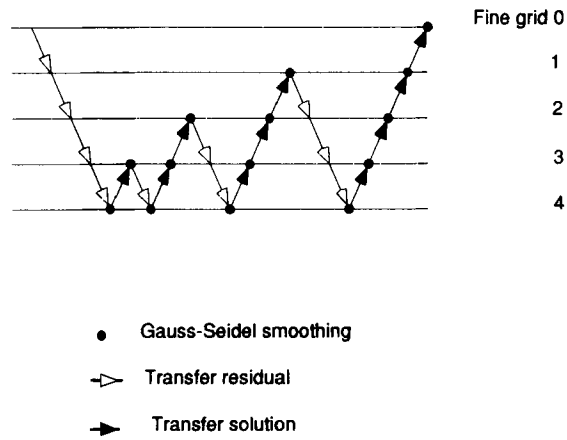


Figure 3. F-cycle strategy for transferring residuals and corrections

As found by Stuben,[12] this improves the convergence.

Smoother

A simple point Gauss–Seidel relaxation is used for smoothing. Just one or two sweeps have been found to be satisfactory. A single sweep on the coarsest grid equation is of course sufficient to give the exact solution.

Relaxation factor

In principle this basic scheme requires no underrelaxation to ensure convergence. In practice a very small amount may be allowed to cover against round-off errors spoiling the desired row sum condition $\Sigma_j A_{ij} \geq 0$ for $A_{ii} > 0$. This is achieved by increasing the diagonal and adding a compensation to the right-hand side b_i of equation (11).

Performance

To illustrate the performance of this basic scheme, results are presented in Figures 4 and 5 for a heat conduction problem in a square box. The computational mesh used is a simple triangulation of a uniform distribution of nodes arranged on a rectangular lattice. Boundary surfaces are all adiabatic, but two corner points (diagonally opposite) are given fixed temperatures. Figure 4 shows that the scaling of computing time with the number of nodes in the discretization is within 4% of the optimum linear scaling. The time includes the grid set-up time.

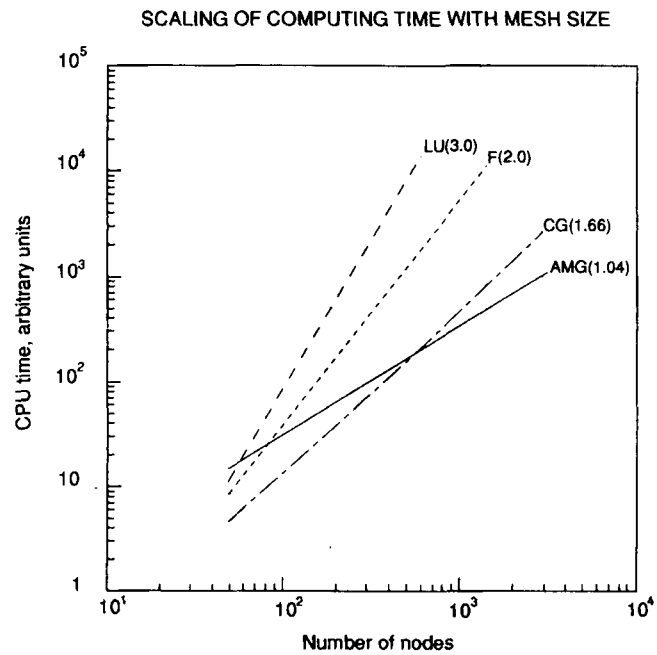


Figure 4. Scaling of linear solvers for conduction in a square box. LU is a direct solver involving LU decomposition with back substitution; F is a direct solver based on the frontal method; CG is a diagonally preconditioned conjugate gradient solver; AMG is the algebraic multigrid solver. Numbers in brackets represents the scaling factor slopes of the curves

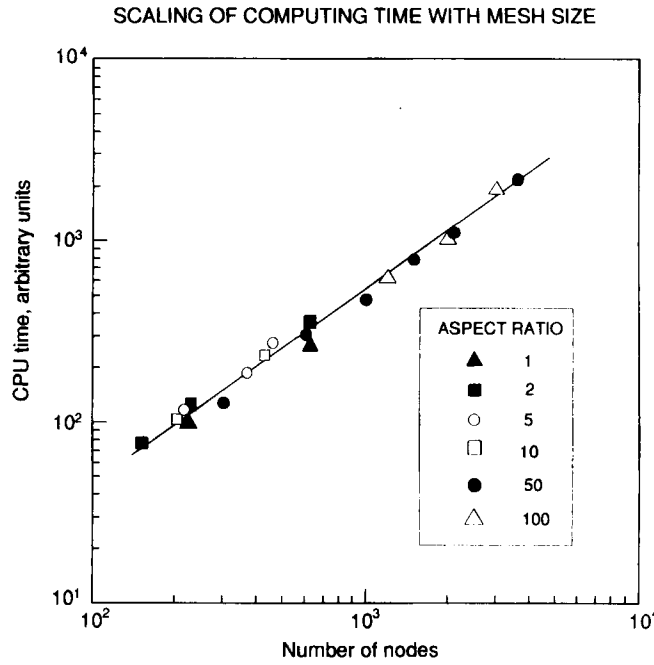


Figure 5. Scaling of the AMG solver for computational meshes of different aspect ratio. The test problem is the same as that for Figure 4

For comparison the scalings of other popular solvers are also shown: LU decomposition with back substitution, the frontal method and a conjugate gradient method with diagonal preconditioning. For discretizations exceeding about 400 nodes the AMG performance is superior.

Figure 5 gives an indication of the mesh independence of the convergence of the solver. The scalings for aspect ratios of nodal spacings in the range 1–100 all lie on the same characteristic.

The conjugate gradient performance, not shown in Figure 5, deteriorated markedly with increasing aspect ratio. Note also that the conjugate gradient solver would not be suitable for the asymmetric matrices associated with the Navier–Stokes equations.

3.2.2. Coupled variable AMG. The scheme for coupled variables is essentially the same as the basic scheme but with some additional constraints on the coarsening algorithm in view of the more complex system matrix. A small change is made to the solution strategy and a provision for relaxation of the coupling between variables is introduced.

Coarsening

In general a system matrix for a coupled variable equation set will consist of diagonal and off-diagonal blocks, the off-diagonal blocks representing the coupling between variables. For mixed vector and scalar variables for example these off-diagonal blocks represent discrete gradient and divergence operators as in equation set (6). The strategy for coarsening such coupled equation sets is to preserve this form of the system matrix down through all the grids. This is equivalent to choosing K to have a block diagonal form, i.e. forbidding the combination of equations for different variables. Note that this does not prevent different coarsening for different variables.

Multigrid cycling strategy

The same F-cycle solution-cycling strategy is used and the same point Gauss–Seidel relaxation is employed on all the grids except the coarsest. Because the coarsening rule forbids the joining of equations for different variables, the coarsest equation set will now be a coupled equation set, so a direct solver is used to obtain an accurate solution.

Direct solvers such as LU decomposition with back substitution are very efficient for small matrices (Figure 4), so no severe penalty is incurred. In fact we can afford to stop the coarsening at a higher level (between about 30 and 60 equations). Note that introducing a direct solver at this level will not degrade the overall scaling. For fine grids the Gauss–Seidel relaxation is applied to the u-, v- and p-equation sets (in that order). For each sweep through the u- and v-equation sets there are two sweeps through the p-equations (first to last and last to first).

Underrelaxation

It has been found necessary to make provision for a small amount of underrelaxation of the coupling between variables. For example, if ω is the relaxation factor for $\mathbf{p}$ and $\mathbf{p}^n$ is the latest iterate, then $\mathbf{p}^n$ is replaced according to

$$\mathbf{p}^n \Rightarrow \omega \mathbf{p}^n + (1 - \omega)\mathbf{p}^{n-1}, \quad \omega \leq 1{\cdot}0, \tag{17}$$

in the equation sets for $\mathbf{v}$, where $\mathbf{p}^{n-1}$ is the previous Navier–Stokes iterate. Similar underrelaxation can be made for $\mathbf{v}$.

In practice very little if any underrelaxation is necessary. Typically values of ω in the range

$$1{\cdot}0 \geqslant \omega \geqslant 0{\cdot}8$$

are sufficient to ensure rapid convergence. The overall convergence of the Navier-Stokes solver is not sensitive to the precise value of ω in this range.

4. PERFORMANCE

4.1. Navier–Stokes convergence characteristics

In this subsection convergence factors are used to quantify convergence rates. The convergence factor is defined as the reduction in the absolute maximum residual per iteration. Thus the smaller the convergence factor, the more rapid is the convergence.

4.1.1. Convergence. Figure 6 shows the implicit AMG solver convergence characteristics for a driven cavity test problem on a uniform mesh at $Re = 100$. It will be evident that convergence rates are largely mesh-independent. If anything, there is slight improvement with mesh refinement. Just 10 iterations are sufficient to take the residual to machine accuracy.

For comparison a convergence characteristic for the SIMPLE algorithm is also shown. This can be described as a segregated solution AMG solver, i.e. algebraic multigrid is used as the linear solver for both velocity and pressure within the SIMPLE algorithm. It will thus give the best performance that could be expected with a segregated solution algorithm for unstructured mesh applications. Nevertheless, on this structured mesh problem its initial convergence factor 0·88, is poor compared with that of the implicit AMG solver, 0·24. Thus after as many as 30 SIMPLE iterations the level of convergence is only just comparable with that of the implicit AMG solver after just three iterations.

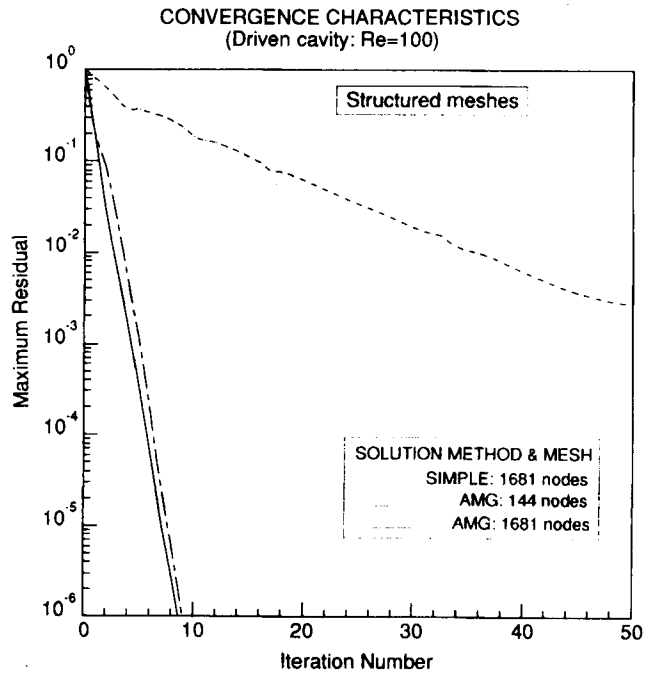


Figure 6. Convergence characteristics for the AMG solver for the dirven cavity problem at $Re = 100$; uniform meshes of 1681 nodes and 144 nodes. For comparison is shown the convergence characteristic for the SIMPLE algorithm on the 1681-node mesh

The initial SIMPLE convergence rate, moreover, is not maintained, as is illustrated in Figure 7 where after about 100 iterations the convergence factor has deteriorated to 0·97. Many hundreds of SIMPLE iterations would be required to achieve machine accuracy.

Figures 8 and 9 show the convergence characteristics for an unstructured mesh of 1509 nodes. For comparison the convergence rate for a uniform mesh of 1681 nodes is also shown. A two-dimensional, 2:1 local mesh refinement is used in the unstructured mesh to even out the spatial variations in mesh Peclet number for this problem (i.e. higher nodal concentrations in regions of high velocity). The implicit AMG solver convergence characteristics are almost identical. For the SIMPLE algorithm, however, the initial convergence factor of 0·88 is maintained for fewer iterations and as a result the overall convergence factor is reduced.

4.1.2. Scaling. The scaling of the convergence rate with problem size has been investigated at Reynolds numbers of 10, 100 and 400 for maximum mesh Peclet numbers in the range 0·25–33·3. Table I gives the convergence factors for the driven cavity problem for different meshes.

For this problem the mesh Peclet number should satisfy

$$Pe \ll Re^{1/2}$$

in order for the boundary layers to be resolved. The condition is satisfied for the $Re = 10$ case and it will be apparent that the convergence factor is largely independent of the number of unknowns. Except for the coarsest mesh in the $Re = 100$ case the condition is also reasonably well satisfied and again the convergence factor is otherwise mesh-independent. At $Re = 400$,

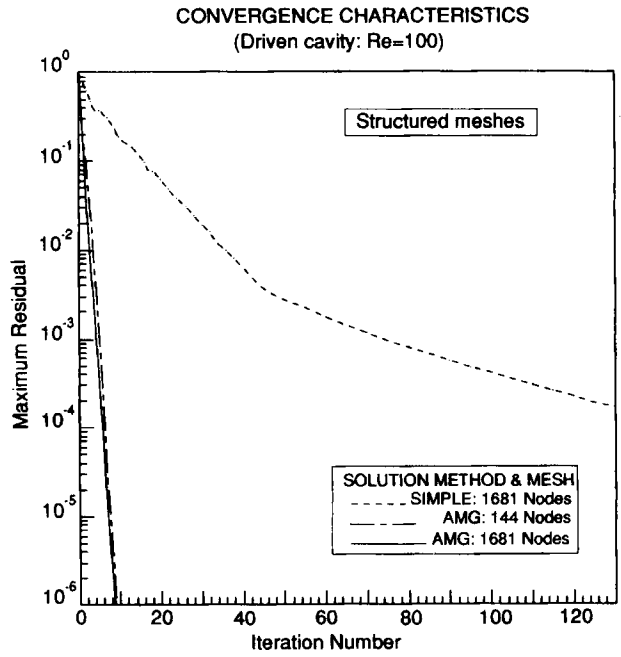


Figure 7. Comparison of the convergence characteristics for the AMG solver and the SIMPLE algorithm; driven cavity at $Re = 100$; uniform meshes of 1681 and 144 nodes

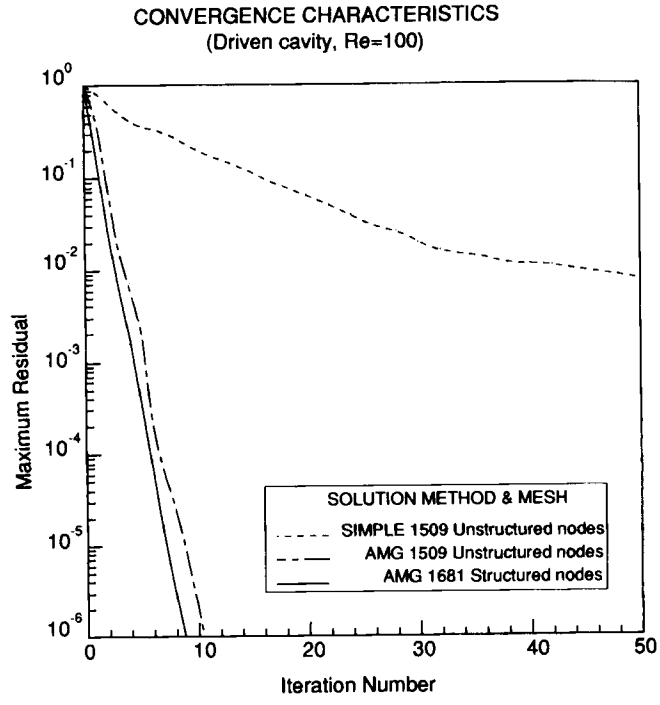


Figure 8. Convergence characteristics of the AMG solver for an unstructured mesh of 1509 nodes with a 2:1 mesh grading. The convergence characteristic for a structured mesh is shown for comparison

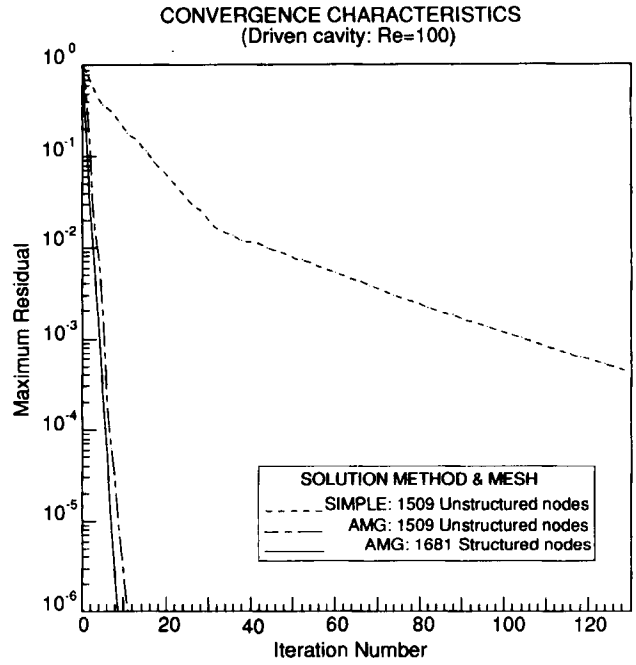


Figure 9. Comparison of the convergence characteristics of the AMG solver and the SIMPLE algorithm; unstructured mesh of 1509 nodes with a 2:1 mesh grading. The convergence characteristic of the structured mesh is shown for comparison.

Table I. Convergence factors for implicit AMG; driven cavity test problem (N, number of unknowns; Pe, mesh Peclet number; α, convergence factor)

			$Re = 10$			
N	243	432	1323	2187	3267	5043
Pe	1·25	0·91	0·5	0·38	0·31	0·25
α	0·064	0·058	0·060	0·062	0·055	0·056
			$Re = 100$			
N	432	1323	2187	3267	5043	
Pe	9·1	5·0	3·8	3·1	2·5	
α	0·31	0·24	0·24	0·24	0·24	
			$Re = 400$			
N	324	1818	2586	4527		
Pe	33·3	13·3	11·1	8·3		
α	0·64	0·56	0·54	0·45		

however, the meshes employed are relatively coarse and the above criterion is barely satisfied; it is actually violated with the coarsest mesh. The convergence factors are then less impressive but nevertheless quite respectable when compared with those for decoupled solution methods even for the coarsest mesh. There is also a marked improvement in the convergence rate with mesh refinement. It is concluded that true multigrid performance and scaling can be achieved with this implicit AMG solver provided that the computational mesh is sufficient to resolve the flow field.

The convergence factors of Table I were obtained for an interpolation scheme which used the complete element as the subcontrol volume. When a more accurate interpolation scheme was used, based on the subcontrol volume illustrated in Figure 2, the convergence factors were not quite as good at low Reynolds numbers. Table II gives typical values, again for Reynolds numbers in the range 10–400 (mesh Peclet numbers in the rantge 0·24–40). These convergence factors are still far superior to those for segregated solution methods, as is their scaling.

4.2. Navier–Stokes test problems

This form of the discrete transport equations is based on the simplified implementation of the Prakash–Hookey scheme. To assess the scheme, it has been applied to a number of standard test problems. The results from just three problems have been selected to illustrate the performance. All the results presented have been obtained with an upwind differencing scheme for advection that is between first- and second-order-accurate. Higher-order schemes have been left for future investigation. The test problems are

(i) a driven cavity
(ii) flow over a backward-facing step
(iii) pipe flow with an internal section of volumetric drag.

The third is a simple test for false compressibility.

Table II. Convergence factors for implicit AMG solver; interpolation based on subcontrol volume of Figure 2 (N, number of unknowns; Pe, mesh Peclet number; α, convergence factor)

		$Re = 10$		
N	363	1323	2883	5043
Pe	1·0	0·5	0·33	0·243
α	0·14	0·14	0·12	0·13
		$Re = 100$		
N	363	1323	2883	5043
Pe	10·1	5·0	3·3	2·43
α	0·30	0·28	0·29	0·28
		$Re = 400$		
N	363	1323	2883	5043
Pe	40·0	20·0	13·3	9·75
α	0·51	0·47	0·46	0·44

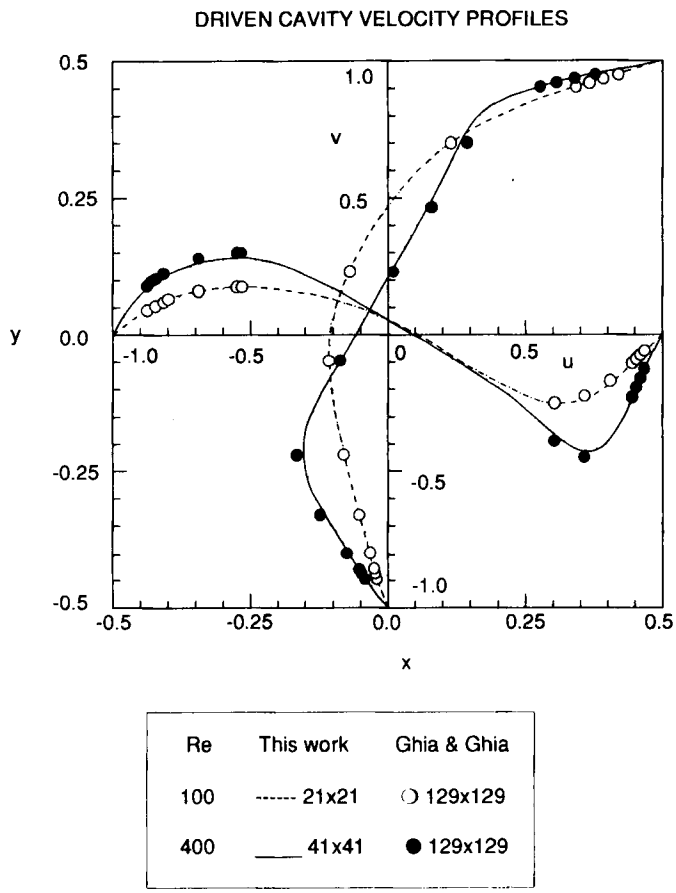


Figure 10. Velocity profiles along vertical and horizontal lines through the centre of a driven cavity at $Re = 100$ and 400. Comparison with the benchmark data of Ghia *et al.*[13]

4.2.1. Driven cavity. Velocity profiles $u(x = 0{\cdot}5, y)$ and $v(x, y = 0{\cdot}5)$ for fluid in a unit-square cavity driven at Reynolds numbers of 100 and 400 ($u(x, y = 1) = 1{\cdot}0$) are shown in Figure 10. Uniform meshes of 21 × 21 and 41 × 41 were used respectively. Superimposed are the benchmark data points of Ghia *et al.*[13] obtained on a non-uniform mesh of 129 × 129. The maximum velocities along the $x = 0{\cdot}5$ axis are within 2% and 3·5% of the benchmarks respectively. These are remarkably accurate predictions bearing in mind the coarseness of the meshes. As noted by Schneider,[14] Galpin[15] found much larger errors using conventional finite difference methods on similar grids.

4.2.2. Flow over a backward-facing step. A fluid enters a two-dimensional channel of width 0·2 m with a parabolic velocity profile, maximum velocity 1·0 m s^{-1}. Some distance down the channel there is a one-sided step increase in width to 0·3 m. A recirculation is set up behind the step. The longitudinal extent of the recirculation is marked by the 'point of reattachment' or the point at which unidirectional flow is re-established across the entire width of the channel.

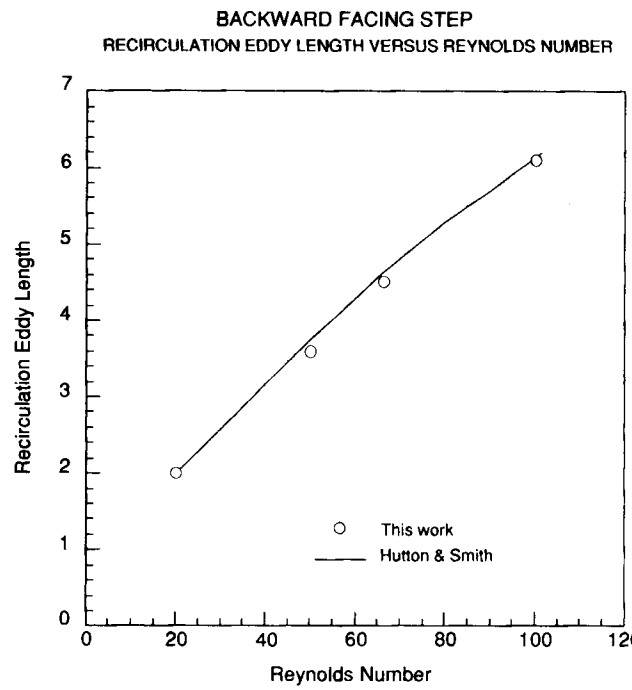


Figure 11. Length of the recirculation eddy for flow over a backward-facing step at Reynolds numbers in the range 20–100 (based on step height). Comparison with the data of Hutton and Smith

Prediction of the point of reattachment and its dependence on Reynolds number is the gauge used to assess the performance of Navier–Stokes solvers for this problem.

In Figure 11 the results obtained for an unstructured mesh of 1331 nodes are compared with the calculations of Hutton and Smith.[16] In both cases the inlet boundary condition was the prescribed parabolic profile, while constant pressure, continuitive outflow was prescribed for the outlet boundary of the calculational domain. The recirculation length is measured in units of step height. Step height is also used as the characteristic length in the definition of Reynolds number, together with the mean channel flow velocity as the characteristic velocity scale.

For computational meshes of this size first-order methods are known to underpredict the reattachment length by as much as 10% or more. The results obtained here are within 3% of accurate solutions produced by Hutton and Smith using a higher-order method.

4.2.3. Pipe flow with internal volumetric resistance. A channel of width 0·4 m and length greater than 1 m contains a resistive section between 0·4 and 0·6 m from the channel entrance. A fluid with unit viscosity and unit density flows through the channel, entering with a parabolic profile and unit peak velocity. The resistance is such as to double the pressure drop down a 1 m length of the pipe, i.e. from 50 to 100 kgf m^{-1} s^{-2}.

Three simulations of this idealized problem have been carried out with three different discretizations of the Navier–Stokes equations, but all based on linear triangular elements. The first, A, is a Galerkin discretization with a segregated method of solution similar to that described by Shaw.[17] The second, B, is similar to A but incorporates the Rhie–Chow approximation; it may be described as the finite element equivalent of the Rhie–Chow method; it considerably

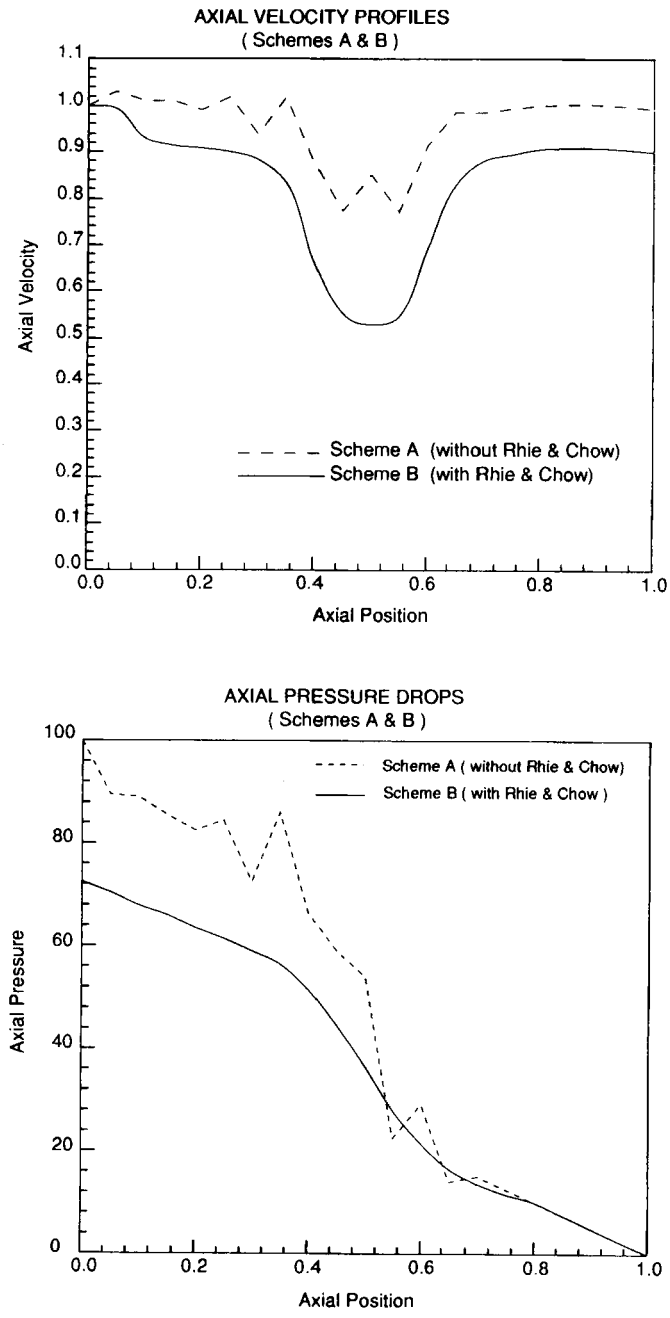


Figure 12. Axial velocity and pressure profiles for flow through a pipe containing a section of volumetric resistance between 0·4 and 0·6 m from the inlet. Predictions for the Galerkin-based discretizations A and B with a segregated solution method

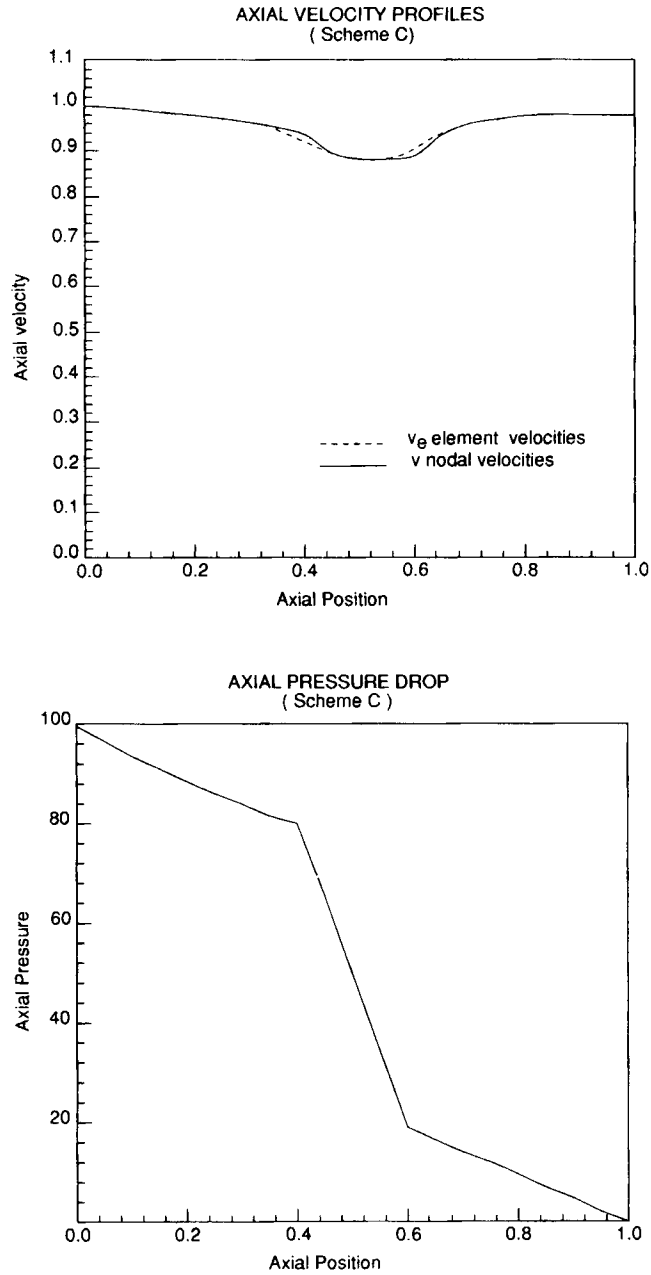


Figure 13. Axial velocity and pressure profiles for flow through a pipe containing a section of volumetric resistance between 0·4 and 0·6 m from the inlet. Predictions for the simplified Prakash–Hookey scheme C with AMG coupled solution

improves the convergence characteristics of method A. Finally, the third method, C, is the simplified Prakash–Hookey scheme of this work.

The axial velocity profiles along the length of the pipe for methods A and B are shown in Figure 12 for a coarse mesh of 9×21 nodes. Method A exhibits 'wiggles' in both pressure and velocity. It displays the correct pressure drop, but the nodal velocities do not satisfy continuity. Method B on the other hand does not exhibit 'wiggles', but the simulated fluid is unable to support the correct pressure drop. This is due to the false compressibility associated with the Rhie–Chow term in the pressure equation. It also results in a large drop in the axial velocity in the resistive zone which exceeds that which could be expected from a redistribution of flow.

In Figure 13 the results for the new scheme, method C, are shown. The results closely follow the correct solution within the discretization error for this coarse mesh. The nodal velocities are very close to the continuitive velocities, the dip in the resistive zone being entirely associated with flow redistribution.

These results demonstrate that this simple implementation of the Prakash–Hookey scheme does not suffer from false compressibility or 'wiggles' and is therefore superior to those schemes based on the Rhie–Chow method of interpolation.

5. GENERAL COMMENTS

The performance characteristics demonstrated in these test problems meet the requirements outlined in Section 1 for addressing the problem of spatial resolution in fluid dynamics. The solution scheme offers

(i) good convergence characteristics
(ii) good scaling of convergence characteristics
(iii) good coupling between flow variables for unstructured finite element meshes.

With close-to-optimum linear scaling the method is also well placed for exploiting the large increases in computing power that will become available through large-scale MIMD parallel processing architectures. Since, moreover, massively parallel MIMD machines are now being produced, the solver could represent a timely development, offering an efficient method of harnessing such computing power to the fluid flow problem.

Other field problems such as those in stress analysis, electromagnetism and magnetohydrodynamics may also benefit from this approach.

6. CONCLUSIONS

An algebraic multigrid algorithm has been produced for solving the transport equations for fluid flow. The discretization gives good coupling between flow variables for unstructured finite element meshes. The solution algorithm is efficient, giving a rapid convergence to machine accuracy that is almost mesh-independent. The scaling of computing time with mesh size is close to the optimum.

ACKNOWLEDGEMENTS

I am indebted to A. Burns and R. D. Lonsdale who first brought by attention to the possibility of a simplified form of the Prakash–Hookey and Schneider–Raw discretizations. I am also grateful for the help and encouragement of G. Robinson, B. R. MacGregor and J. Keeton.

REFERENCES

1. J. Linden, G. Londsdale, B. Steckel and K. Stuben, *Arbeitspapiere der GMD 322*, 1988.
2. G. Robinson, 'Parallel computational fluid dynamics on unstructured meshes using algebraic multigrid', *Proc. Parallel CFD 92 Conf. on Implementations and Results Using Parallel Computers*, Rutgers University, New Brunswick, May 1992.
3. R. D. Lonsdale, *Proc. Int. Conf. on Numerical Methods in Turbulent and Laminar Flow*, Stanford CA, 1991, pp. 1432–1442.
4. F. Sotiropoulos and S. Abdallah, 'The discrete continuity equation in primitive variable solutions of incompressible flow', *J. Comput. Phys.*, **95**, 212–227 (1991).
5. C. M. Rhie and W. L. Chow, 'Numerical study of the turbulent flow past an airfoil with trailing edge separation', *AIAA J.*, **21**, 1525–1532 (1983).
6. B. R. Baliga and S. V. Patankar, in W. J. Minkowycz *et al.* (eds), *Handbook of Numerical Heat Transfer*, Wiley, New York, 1988, pp. 421–461.
7. C. Prakash, 'An improved control volume finite element method for heat and mass transfer, and for fluid flow using equal-order velocity–pressure interpolation', *Numer. Heat Transfer*, **9**, 253–276 (1986).
8. N. A. Hookey, 'Evaluation and enhancements of control-volume finite-element methods for two-dimensional fluid flow and heat transfer', *M. Eng. Thesis*, Department of Mechanical Engineering, McGill University, Montreal, 1986.
9. G. E. Schneider and M. J. Raw, 'Control volume finite-element method for heat transfer and fluid flow using colocated variables', *Numer. Heat Transfer*, **11**, 363–390 (1987).
10. J. W. Ruge and K. Stuben, 'Algebraic multigrid', in S. McCormick (ed.), *Frontiers in Applied Mathematics*, Vol. 5, *Multigrid Methods*, SIAM, Philadelphia, PA, 1987, pp. 73–130.
11. B. R. MacGregor, unpublished work, 1989.
12. K. Stuben, 'Algebraic multigrid (AMG): experience and comparisons', *Appl. Math. Comput.*, **13**, 419–452 (1983).
13. U. Ghia, K. N. Ghia and C. T. Shin, 'High-*Re* solutions for incompressible fluid flow using the Navier Stokes equations and a multigrid method', *J. Comput. Phys.*, **48**, 387–411 (1982).
14. G. E. Schneider, in W. J. Minkowycz *et al.* (eds), *Handbook of Numerical Heat Transfer*, Wiley, New York, 1988, p. 417.
15. P. F. Galpin, 'Solution of non-linear algebraic equations in the numerical prediction of fluid flow and heat transfer', *M.Sc. Thesis*, University of Waterloo, 1985.
16. A. Hutton and R. Smith, 'The prediction of laminar flow over a downstream facing step by the finite-element method', *CEGB Rep. RD/B/N3660*, Berkeley, 1979.
17. C. T. Shaw, 'Using a segregated finite-element scheme to solve the incompressible Navier Stokes equations', *Int. j. numer. methods fluids*, 12, 81–92 (1991).