INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids **28**: 317–336 (1998)

EFFICIENT ALGEBRAIC MULTIGRID SOLVERS WITH ELEMENTARY RESTRICTION AND PROLONGATION

R. WEBSTER*

Roadside, Harpsdale, Halkirk, Caithness, KW12 6UN, UK

SUMMARY

An algebraic multigrid (AMG) scheme is presented for the efficient solution of large systems of coupled algebraic equations involving second-order discrete differentials. It is based on elementary (zero-order) intergrid transfer operators but exhibits convergence rates that are independent of the system bandwidth. Inconsistencies in the coarse-grid approximation are minimised using a global scaling approximation which requires no explicit geometrical information. Residual components of the error spectrum that remain poorly represented in the coarse-grid approximations are reduced by exploiting Krylof subspace methods. The scheme represents a robust, simple and cost-effective approach to the problem of slowly converging eigenmodes when low-order prolongation and restriction operators are used in multigrid algorithms. The algorithm investigated here uses a generalised conjugate residual (GCR) accelerator; it might also be described as an AMG preconditioned GCR method. It is applied to two test problems, one based on a solution of a discrete Poisson-type equation for nodal pressures in a pipe network, the other based on coupled solutions to the discrete Navier–Stokes equations for flows and pressures in a driven cavity.

KEY WORDS: algebraic multigrid; elementary restriction/prolongation; minimum residual methods

1. INTRODUCTION

The solution of the very large sets of coupled algebraic equations that characterise wide-bandwidth systems, can burden the most powerful computers. Indeed, such calculations are always likely to be resource limited, since the required processing increases disproportionately with the system bandwidth Q, and there is virtually no limit to the range of Q that can be usefully exploited in the representation of natural systems. Efficient solvers are therefore essential. Ideally, solver efficiency (the average number of operations per equation) should be independent of Q. This rules out direct methods such as Gaussian-elimination (GE) and iterative methods such as Gauss–Seidel relaxation (GS) where the efficiency scales as Q^{α}, $\alpha = 2$ (GS) and $\alpha = 2d$ (GE) for systems of topological dimension d. Krylof-subspace and multigrid (MG) accelerated relaxation methods scale better, but of the two, wide-bandwidth MG methods offer the best prospect because convergence rates can be independent of Q, in which case the total operation count for a fixed level of convergence would scale linearly with the total number of equations to be solved, the optimum scaling.

The systems of interest can be expressed as

* Correspondence to: Roadside, Harpsdale, Halkirk, Caithness, KW12 6UN, UK.

CCC 0271–2091/98/120317–20$17.50
Received July 1997
Revised August 1997

$$\mathbf{A}\mathbf{u} = \mathbf{b}, \tag{1}$$

where $\mathbf{A}$ is the system matrix, $\mathbf{u}$ represents the set of unknowns and $\mathbf{b}$ the source vector. As a discrete approximation of a continuum field problem, it could have been derived, e.g. by finite element methods. As an inherently discrete system, it may represent a wide range of network problems in applications as diverse as fluidics, electronics, mechanics, cybernetics, economics etc. Any such system may be represented graphically as a connected nodal network, or grid, with a one-to-one correspondence between equations and nodes of the grid, the coupling between equations being represented by the connections between nodes. The bandwidth of the system may be loosely defined as the 'linear grid size', or

$$Q \sim N^{1/d}, \tag{2}$$

for a grid of N nodes. Clearly, for problems with an explicit geometrical foundation such as continuum fields in a discrete approximation, Q will be related to the resolving power of the discretisation. Therefore, in this case, Q will be used for both system bandwidth and resolving power.

The operator $\mathbf{A}$ may be asymmetric, but for iterative solutions it must be positive and at least semi-definite. In the case of non-linear problems $\mathbf{A}$ will be a linearised approximation that is updated repeatedly at each stage of a Picard or a Newton iteration sequence.

An iterative solution of the linear problem defined by Equation (1) begins with an initial guess $\mathbf{u}^0$, and an associated residual error $\mathbf{r}^0$, given by

$$\mathbf{r}^0 = \mathbf{b} - \mathbf{A}\mathbf{u}^0. \tag{3}$$

A correction, $\mathbf{v}$ to $\mathbf{u}^0$, is sought with a view to eliminating this error. Subtracting Equation (1) from (3) gives the equation to be solved for the correction $\mathbf{v}$:

$$\mathbf{A}\mathbf{v} = \mathbf{A}(\mathbf{u} - \mathbf{u}^0) = \mathbf{r}^0. \tag{4}$$

Approximate solutions for $\mathbf{v}$ are thus sought within the framework of a convergent iterative scheme. At the nth iteration the correction equation will be

$$\mathbf{A}\mathbf{v}^n = \mathbf{r}^{n-1}, \tag{5}$$

and the solution algorithm might take the form

$n = 0$; $\mathbf{r}^0 = \mathbf{b} - \mathbf{A}\mathbf{u}^0$
while ($\|\mathbf{r}^n\| > \varepsilon$ do
  $n = n + 1$
  $\mathbf{d}^n = \Gamma\mathbf{r}^{n-1}$
  $\mathbf{u}^n = \mathbf{u}^{n-1} + \mathbf{d}^n$
  $\mathbf{r}^n = \mathbf{r}^{n-1} - \mathbf{A}\mathbf{d}^n$

The control parameter ε represents the convergence tolerance level. The step $\mathbf{d}^n = \Gamma\mathbf{r}^{n-1}$ represents the approximate solution of Equation (5) using the chosen solution procedure. For example, for relaxation procedures where matrix $\mathbf{A}$ is split according to $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U}$, where $\mathbf{L}$ is the lower triangular block, $\mathbf{D}$ the diagonal and $\mathbf{U}$ the upper triangular block, then $\Gamma = \mathbf{D}^{-1}$ for the Jacobi method, whilst $\Gamma = (\mathbf{L} + \mathbf{D})^{-1}$ for Gauss–Seidel relaxation. The problem with such local relaxation methods is that $\mathbf{d}^n$ is a poor approximation to $\mathbf{v}^n$. As Γ is a short-range operator, the effective bandwidth of $\mathbf{d}^n$ is small compared with that of $\mathbf{v}^n$. Within any single iteration it only effectively addresses the highest-wavenumber domain of the error spectrum, and its effectiveness decreases quadratically with the wavenumber. Convergence is

thus bandwidth dependent, and very inefficient for broad-band, high-resolution, systems. Multigrid methods seek to obtain a full-bandwidth approximation for $\mathbf{v}^n$, and thus bandwidth-independent convergence rates. Such convergence is described as being mesh independent in finite element applications.

The multigrid procedure begins with the usual narrow-band, high-wavenumber approximation $\mathbf{d}^n$, obtained by applying Γ to $\mathbf{r}^{n-1}$ for the reference equation set (fine grid). Corrections at lower wavenumbers are then sought from a reduced-bandwidth (coarse grid) representation obtained by a suitable restriction procedure. In algebraic multigrid (AMG) methods the procedure is algebraic. Thus, if $\mathbf{K}$ is the restriction operator, applying it to Equation (5) will generate a smaller equation set (coarser grid):

$$\mathbf{A}^c \varphi^c = \mathbf{r}^c, \tag{7}$$

where

$$\mathbf{A}^c = \mathbf{K}\mathbf{A}\mathbf{K}^T, \tag{8}$$

and if $\mathbf{r}^c$ is derived on the basis of the current residual error, i.e.

$$\mathbf{r}^c = \mathbf{K}(\mathbf{r}^{n-1} - \mathbf{A}\mathbf{d}^n), \tag{9}$$

then a solution of Equation (7) provides a lower wavenumber correction, φ^c, that can be used to improve $\mathbf{d}^n$ by increasing its bandwidth, i.e.

$$\mathbf{d}^n \leftarrow \mathbf{d}^n + \mathbf{K}^T \varphi^c, \tag{10}$$

where $\mathbf{K}^T$ is the prolongation operator. Note that $\mathbf{K}^T$ need not necessarily be the transpose of $\mathbf{K}$. Clearly, Equation (7) has the same form as Equation (5), so the procedure can be applied recursively to generate successively smaller equation sets (coarser grids) and successively lower wavenumber corrections. A suitable, cyclic, recursion scheme may then be used to transform φ^c into a wide-bandwidth correction, and hence $\mathbf{d}^n$ into a full-bandwidth approximation for $\mathbf{v}^n$.

The simplest scheme is based on the so-called V-cycle, the restriction of residuals and the prolongation of corrections constituting, respectively, the 'downward' and the 'upward' legs of the cycle. Prior to each restriction, equation sets may be preconditioned with v_1 applications of a short-range relaxation operator, to smooth out the high-wavenumber errors for that grid. Following each prolongation, v_2 applications of a similar smoothing operator will also remove those high-wavenumber errors in the correction that have been generated by the prolongation. Such a V-cycle is commonly designated $V(v_2, v_1)$. Other more elaborate schemes are in general use, such as the W-cycle and the F-cycle. The F-cycle is essentially a V-cycle with an upward leg that is itself comprised of nested V-cycles. Following the usual convention, the notation $F(v_2, v_1)$ will be used below to represent an F-cycle that employs v_1 sweeps of a prerestriction smoother and v_2 sweeps of a postprolongation smoother.

Essential to the achievement of the ideal MG performance are consistent coarse-grid approximations to the problem at each reduced scale of resolution. If there is a breakdown in consistency at any level, then the correction assembled will not make a full-bandwidth improvement. To ensure consistency it is important to choose the restriction and prolongation operators, $\mathbf{K}$ and $\mathbf{K}^T$ carefully. Hemker [1] has shown that the criterion due to Brandt [2] and Hackbush [3] needs to be respected. Thus, if m_r and m_p are the lowest order polynomials (plus one) that are faithfully interpolated by $\mathbf{K}$ and $\mathbf{K}^T$, and m is the order of the partial differential equation being approximated, then

$$m_r + m_p > m, \tag{11}$$

must be satisfied for mesh independent convergence. Thus for second-order partial differential equations (such as Poisson equations), one of the transfer operators must be at least capable of linear interpolation. On the other hand, for first-order equations (such as Euler equations) zero-order operators should suffice. The simplest zero-order restriction procedure is straightforward reduction by the addition of selected equations, and the corresponding prolongation is direct assignment of coarse-grid nodal corrections to contributing fine-grid nodes. These are appealingly simple procedures and especially attractive in view of their low computational cost, both in terms of operation count and storage. Obviously, operators $\mathbf{K}$ and $\mathbf{K}^T$ are implicit to the procedures and need not be formed explicitly. If required they can be assembled by addition of appropriate rows/columns of the unit matrix, which is just the formal definition of an elementary matrix. For this reason reference is made here to 'elementary AMG' (i.e. AMG based on elementary restriction and prolongation operators).

Many implementations of AMG solvers for second-order equations have not fully respected condition (11). For example, in the development of the AMG method, Ruge and Stuben [4] used matrix-dependent operators which did not strictly comply with the criterion, but they found that the grid dependence was relatively weak. They also found that improved interpolation procedures, which virtually eliminated the mesh dependence, were not cost effective for the range of test problems tried. Other authors have used simpler low-order operators for second-order equations. Thus the control-volume agglomeration method of Lallemand *et al.* [5] and Koobus *et al.* [6] (a special case of AMG) uses elementary operators for restriction and scaled elementary operators for prolongation. Lonsdale [7] and Webster [8–10] used similar operators in AMG linear solvers for Navier–Stokes equations, and both of the latter authors show that, despite not strictly satisfying the criterion, the convergence is not strongly mesh dependent.

Here the linear solver introduced in References [8–10] is modified to improve its effective bandwidth. The modifications represent simple, cost-effective, methods for improving the performance of AMG solvers based on elementary restriction and prolongation. Attention is initially focused on equations with discrete, symmetric, (Poisson-type) operators on a single field variable since it is second-order difference operators that are not considered well served by zero-order interpolation. Thereafter, attention returns to the more complex system of coupled Navier–Stokes equations in the discrete formulation first introduced in Reference [8].

2. SOLUTION METHOD

Descriptions of the elementary AMG solvers to be modified will be found in the quoted references, for a single field variable in Reference [7], for coupled vector and scalar field variables in [8–10]. Only modifications to these basic schemes will be discussed here.

The motivation for retaining elementary intergrid transfer operators is their simplicity and low computational cost. Ruge and Stuben's work suggests that the savings can outweigh the penalty of not strictly satisfying Equation (11), except possibly for problems of very large bandwidth. Moreover, in the context of AMG, resorting to higher order transfer operators could prove difficult; the algebraic coarsening may not always generate coarse grids of the quality required for an homogeneous, isotropic higher order interpolation, in which case there would still be inconsistencies. On the other hand, the adoption of elementary intergrid transfer operators will certainly result in coarse-grid inconsistencies and hence slowly convergent modes. To retain elementary operators and to have efficient reduction factors for all spectral components obviously demands additional measures that address the problem of the slowly

convergent modes, and of course these must not undermine the computational cost advantages of the elementary operator scheme. Two approaches that go some way towards satisfying these requirements have been investigated. The first seeks to minimise the actual number of slowly convergent modes by exploiting some simple techniques for reducing the degree of inconsistency in the coarse-grid approximation. The second seeks to speed up the convergence of those remaining by exploiting well established acceleration techniques. Conjugate gradient accelerators have been shown to be very efficient in speeding up otherwise weakly converging multigrid algorithms [11]. Here the effectiveness of a generalised conjugate residual (GCR) accelerator is investigated.

2.1. The reduction of inconsistencies in the coarse-grid approximation

Some reduction in the number of errant modes can be achieved by not pursuing coarsening to the lowest resolution. By terminating at a higher level and using an alternative solver for the larger coarsest grid, inconsistencies associated with lower levels are avoided [8]. The coarsest grid is chosen to be as large as possible, subject to the requirement that the alternative solver is competitive with AMG. As direct methods (such as GE) are competitive for coarse grids, a reduction, though a somewhat limited one, is possible.

The inconsistencies at higher levels must be tackled directly. Consider the following finite element field problem as a heuristic. If the computational domain is discretised to a resolved length scale λ, then matrix entries for a discrete, second-order, partial differential operator, $\mathbf{A}$, should scale as λ^{-2}. Therefore, if the fine grid is reduced to a lower resolution, λ_c, by a suitable coarsening procedure (restriction and prolongation operators $\mathbf{R}$ and $\mathbf{P}$ respectively), then matrix entries for the coarse-grid approximation, $\mathbf{A}_c = \mathbf{RAP}$, should be proportional to λ_c^{-2}. Elementary restriction and prolongation, $\mathbf{R} = \mathbf{K}$, $\mathbf{P} = \mathbf{K}^T$, fails to satisfy condition (11) and is thus unable to provide this consistency. Matrix entries for $\mathbf{A}_c = \mathbf{KAK}^T$ are actually proportional to $(\lambda_c \lambda)^{-1}$. To recover consistency it is necessary to apply a scaling factor $\sigma = \lambda/\lambda_c$ at each stage of the coarsening procedure. For a one-dimensional uniform discretisation, uniformly coarsened, the factor is easily deduced to be 0.5 [4]. For multi-dimensional problems discretised on the basis of unstructured finite element meshes, the required scaling is less easily deduced without recourse to computationally expensive measures. Koobus *et al.* [6] proposed an empirically derived factor which is exact for certain special cases and which should at least reduce the inconsistency in others. In the present study, only an approximate scaling, σ, the target ratio of the bandwidths of successive grids will be used, or

$$\sigma = (\beta)^{1/d}, \tag{12}$$

where β is the reduction factor for grid coarsening and d is the topological dimension of the fine grid. Since β is a global coarsening parameter, the approximation will be called the global scaling approximation. Note that it involves the recursive application of σ down through the entire coarse-grid hierarchy. Matrix coefficients for grid level n will all be reduced by a factor, $\beta^{(n-1)/d}$, compared with those without such scaling. Note also that σ uses no explicit geometrical information, so its use is not necessarily restricted to the chosen heuristic.

2.2. Acceleration of the slowly convergent modes

The basic algorithm, without an accelerator, will be referred to as algorithm A. It is essentially the same as iterative

Algorithm A

$n = 0;\ \mathbf{r}^0 = \mathbf{b} - \mathbf{A}\mathbf{u}^0$
while $(\|\mathbf{r}^n\| > \varepsilon)$ do
  $n = n + 1$
  $\mathbf{d}^n = \mathbf{F}(v_2,\ v_1)\mathbf{r}^{n-1}$
  $\mathbf{z}^n = \mathbf{A}\mathbf{d}^n$
  $\mathbf{u}^n = \mathbf{u}^{n-1} + \mathbf{d}^n$
  $\mathbf{r}^n = \mathbf{r}^{n-1} - \mathbf{z}^n$

solution algorithm (6), with Γ replaced by the multigrid operator for elementary AMG. The step $\mathbf{d}^n = \mathbf{F}(v_2, v_1)\ \mathbf{r}^{n-1}$, thus represents the approximate solution of Equation (5) using a single F-cycle. Contraction operators $\hat{\mathbf{C}}$ and $\mathbf{C}$ relating successive residual and solution errors are

$$\hat{\mathbf{C}} = \mathbf{A}\mathbf{C}\mathbf{A}^{-1} = \mathbf{I} - \mathbf{A}\mathbf{F}(v_2, v_1). \tag{13}$$

The eigenvalues of $\mathbf{AF}(v_2, v_1)$ should be highly clustered and very close to unity. Only a small number associated with the errant modes should remain distinct, which is an ideal preconditioning for a conjugate gradient accelerator. Since $\mathbf{A}$ is not necessarily symmetric, this would need to be an unsymmetric conjugate gradient accelerator. Thus, in algorithm B the solver is driven with a GCR control harness. Successive search vectors are made orthogonal (both $\mathbf{d}^n$ and $\mathbf{z}^n$) using a Gramm–Schmidt orthonormalisation and the corrections applied are scaled to minimise residual norms with respect to the current and all previous search directions. An additional preconditioning, $\hat{\mathbf{C}}(\mathbf{b} - \mathbf{A}\mathbf{u}^0)$, is introduced together with a restart capability, the control parameter η being an upper limit to the number of allowed iterations before each restart. The shorthand GCR(η) is frequently used for this type of accelerator [12]. The disadvantage with this approach is of course the extra

Alogrithm B

while $(\|\mathbf{r}^n\| \geq \varepsilon)$ do
  $n = 0;\ \mathbf{r}^0 = \hat{\mathbf{C}}(v_2,\ v_1)(\mathbf{b} - \mathbf{A}\mathbf{u}^0)$
  while $(n < \eta$ and $\|\mathbf{r}^n\| > \varepsilon)$ do
  $n = n + 1$
    $\mathbf{d}^n = \mathbf{F}(v_2,\ v_1)\mathbf{r}^{n-1}$
    $\mathbf{z}^n = \mathbf{A}\mathbf{d}^n$
    for $(i = 1,\ n - 1)$ do
      $\alpha^{(n,\,i)} = \langle \mathbf{z}^n, \mathbf{z}^i \rangle / \langle \mathbf{z}^i, \mathbf{z}^i \rangle$
      $\mathbf{z}^n = \mathbf{z}^n - \alpha^{(n,\,i)}\mathbf{z}^i$
      $\mathbf{d}^n = \mathbf{d}^n - \alpha^{(n,\,i)}\mathbf{d}^i$
    $\omega = \langle \mathbf{r}^{n-1}, \mathbf{z}^n \rangle / \langle \mathbf{z}^n, \mathbf{z}^n \rangle$
    $\mathbf{u}^n = \mathbf{u}^{n-1} + \omega\mathbf{d}^n$
    $\mathbf{r}^n = \mathbf{r}^{n-1} - \omega\mathbf{z}^n$
  $\mathbf{u}^0 = \mathbf{u}^n$

storage required, which increases with the iteration count to the limit set by η. Alternatives that may require less memory are the biconjugate gradient (BICG) algorithm [13] and the conjugate gradient squared (CGS) algorithm [14]. However, providing the preconditioner is effective, the number of distinct clusters of eigenvalues, say n_c, should be small, and the algorithm should converge in about n_c iterations. Therefore, the storage requirements should not be prohibitive, $\eta \leq n_c$, and may be comparable with those for BICG and CGS. It is

estimated that η could be as large as 2.5 times the average nodal connectivity before algorithm B would begin to show any storage disadvantage when compared with alternative schemes based on higher order interpolation (assuming the latter do not require an accelerator).

In the following, the improvements have been applied sequentially. Thus, in ascending order of improvement, reference is made to algorithms A, A′ and B′, where primed identifiers indicate that the global scaling approximation has also been used.

3. NUMERICAL RESULTS

Before discussing the results for the test applications, Darcy flow in pipe networks and Navier–Stokes flow in a square cavity, it will be helpful to specify the solution parameters used, and to define the assessment parameters.

3.1. Solution method parameters

For Darcy flows, an F(1, 1)-cycle schedule is used with the grid hierarchy extending down to a single equation. For Navier–Stokes flows an F(3, 1) schedule is used, the grid hierarchy being terminated at about 20 equations. A GCR solver provides an accurate solution for this coarse grid set. The postsmoothing is also driven with a GCR harness wherever more than one smoother sweep is employed. Prior to prolongation, corrections are also scaled, as described in Reference [10]. No fixed limit is placed on the number of F-cycles; unless otherwise indicated, residuals are reduced to the level of machine accuracy.

3.2. Aspects of performance assessment

3.2.1. Convergence characteristics. When residual error norms are plotted logarithmically against the iteration count, a convergence characteristic is obtained. For the convergence characteristics shown here, residual norms have been normalised by the initial Euclidean norm. Note that in Section 3.3 this is the norm of the residual ready for the first restriction, i.e. *after* preconditioning with v_1 sweeps of the Gauss–Seidel smoother. Total error reductions are thus somewhat larger than those suggested by the figures shown, the reduction of the initial high-wavenumber errors being effectively discounted in the assessment of performance.

3.2.2. Residual reduction factors. The slope between successive points will be used to derive residual reduction factors for each characteristic. Where convergence characteristics cannot be characterised adequately by a single, average, reduction factor, two selected reduction factors are used; the maximum (i.e. the worst) reduction factor and the average reduction factor, $\bar{\mu}$. For a sequence of n F-cycles $\bar{\mu}$ is defined as

$$\bar{\mu} = \{\|\mathbf{r}^{(n)}\|_2/\|\mathbf{r}^{(0)}\|_2\}^{1/n} = \{\Pi_0^n \mu^{(i)}\}^{1/n}, \tag{14}$$

where μ_i, the reduction factor for the ith F-cycle, is given by

$$\mu^{(i)} = \|\mathbf{r}^{(i)}\|_2/\|\mathbf{r}^{(i-1)}\|_2, \tag{15}$$

where $\mathbf{r}^{(i)}$ is the residual following the ith cycle. Note that the F-cycle is used to define the reduction factors for both algorithms A and B.

3.3. Application 1: Darcy flow in pipe networks

3.3.1. Test problems. These test problems are based on laminar pressure-driven flow in pipe networks. Pipes are straight and sufficiently long for junction pressure losses to be ignored.

The task is essentially the solution of the continuity equation, which takes the form of a discrete Poisson-like equation for the nodal (pipe junction) pressures. Two well-separated nodes in the networks are prescribed different pressures, all other nodes are free. The problem is thus the determination of the free node pressures and hence the pipe flows for the entire network. Two types of network have been studied:

Type 1. Uniform networks in one and two dimensions with linear nodal connectivity, $\chi \leq 3$, and rectangular nodal connectivity, $\chi \leq 5$.

Type 2. Unstructured pipe networks within a rectangular domain with variable nodal connectivity, $\chi \leq 10$.

Simple examples are illustrated in Figure 1. In both two-dimensional cases the domain is rectangular and the two fixed nodes have been chosen to be those at diagonally opposite corners of the network. The pressure drop is chosen to ensure laminar flow with the maximum Reynolds number, *Re*, not exceeding 10^3.

3.3.2. Results. Examples of the convergence characteristics obtained for each solution method when applied to type 1 problems in one-dimension are given in Figure 2 for bandwidths spanning four orders. The broken curves are for algorithm A, elementary AMG, without any scaling of the coarse-grid approximation. Clearly, multigrid performance breaks down at the higher bandwidths (500 and above) with maximum reduction factors approaching or exceeding unity. This is the grid-dependent behaviour expected for an inconsistent coarse-grid approximation.

For this simple one-dimensional case, the coarse-grid approximation is greatly improved by scaling alone. Convergent behaviour is then obtained for all orders of bandwidth, the chain lines (A′) in Figure 2. Note, however, that there is still evidence of a weak bandwidth dependence, the number of iterations required to achieve a six-order reduction in the residual norm increasing by about one iteration for every order of magnitude increase in bandwidth. This indicates that there is still some inconsistency which leaves a few eigenmodes slowly convergent. The GCR(10) accelerator deals effectively with these, bringing the number of iterations required to ten in all cases (the points linked by full lines (B′) in Figure 2).

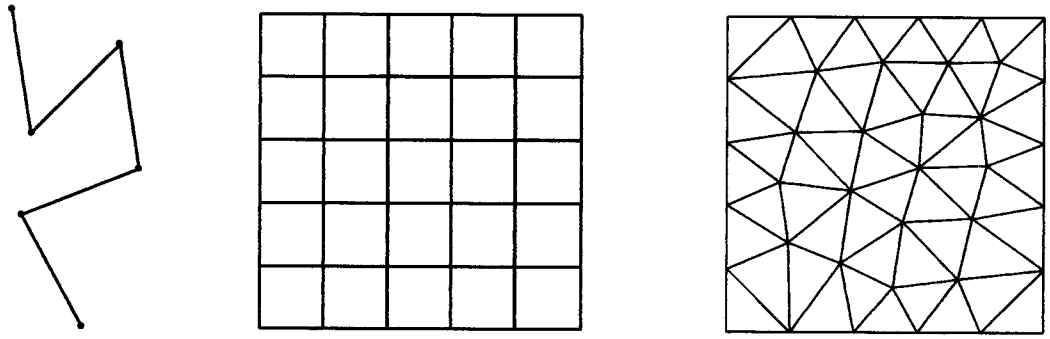

Figure 1. Simple examples of structured and unstructured pipe networks. (a) Type 1 ($\chi \leq 3$, $Q = 5$; five pipes, six nodes). (b) Type 1 ($\chi \leq 5$, $Q = 5$; 60 pipes, 36 nodes). (c) Type 2 ($\chi \leq 8$, $Q = 7$; 91 pipes, 38 nodes).

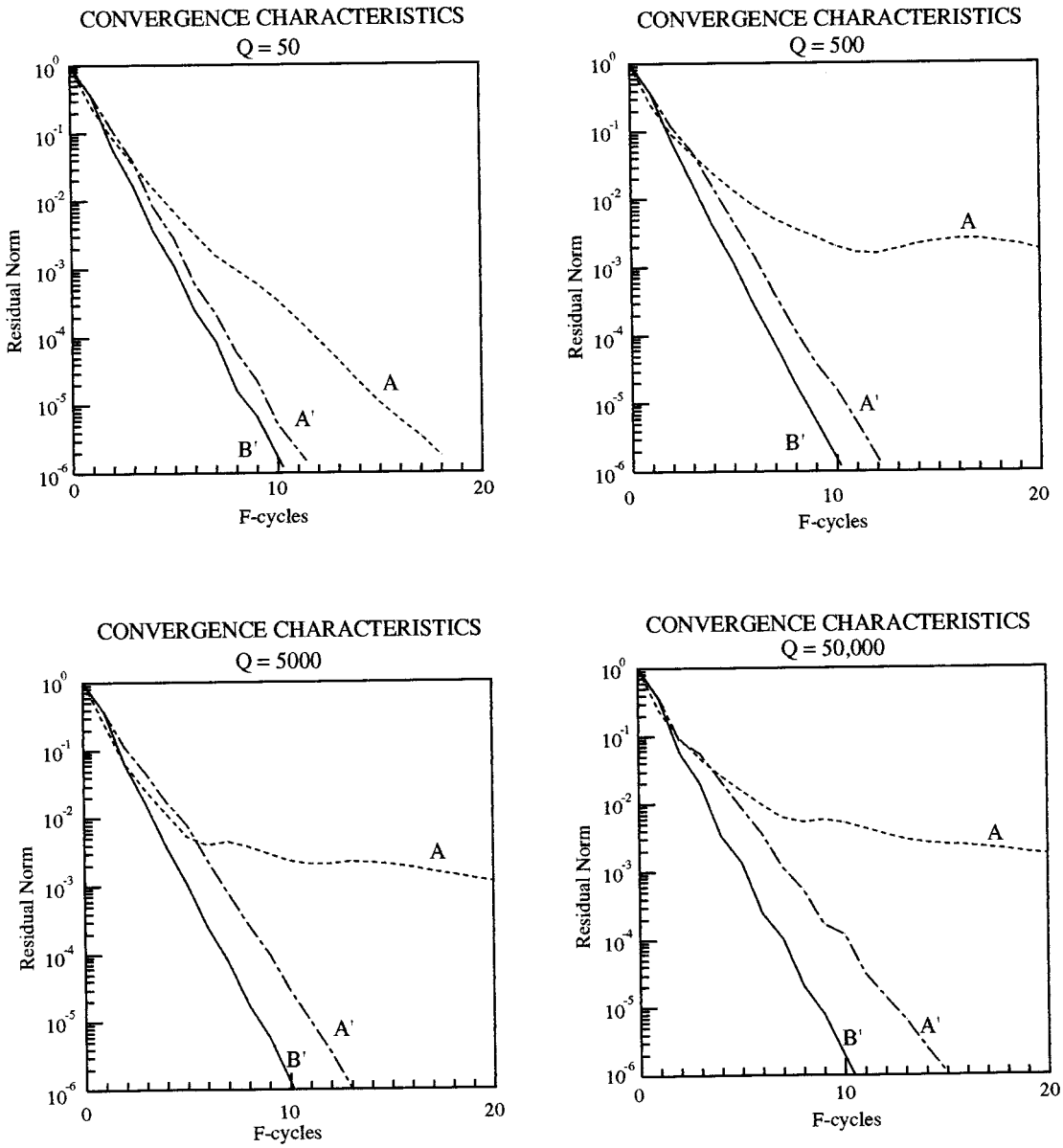

Figure 2. Convergence characteristics for Darcy flow in one-dimensional, type-1, networks; A (elementary AMG), A′ (elementary AMG with scaling of coarse-grid equations) and B′ (GCR-accelerated AMG with scaling of coarse-grid equations).

For multi-dimensional networks and non-uniform, unstructured, networks the global scaling factor σ is unlikely to provide such a good coarse-grid approximation. This is clearly evident in Figure 3, which gives the convergence characteristics for two-dimensional meshes with bandwidths of $Q = 130$ and 219 for uniform and non-uniform cases respectively. In both cases algorithm A′ is less satisfactory, the convergence being sluggish and erratic (points joined by the chain line, A′). For the unstructured network, reduction factors actually exceed unity at

some points, though convergence is recovered in subsequent iterations. In contrast, the efficiency of algorithm B′ is undiminished (points joined by the full lines, B′) The GCR accelerator has little difficulty in dealing with the slowly convergent modes; convergence is just as good as that for the one-dimensional cases. The convergence rate is maintained down to machine accuracy, (Figure 4). Moreover, it is not sensitive to the particular GCR(η) restart schedule adopted, indicating that, at least for these symmetric cases, the storage requirement can indeed be minimised if necessary by choosing small values for η without degrading the performance significantly.

Further results for algorithms A′ and B′ for two-dimensional networks of both types are summarised in Tables I, II, III and IV. These give maximum reduction factors, mean reduction factors and the number of F-cycles required to reduce residual norms by six orders, for bandwidths spanning $Q = 29–290$ (uniform networks Tables I and II) and $Q = 32–323$ (non-uniform networks Tables III and IV). They show again that without the GCR accelerator performance degrades at large bandwidths, reduction factors straying on occasion into divergent values. Reduction factors for the accelerated algorithm show no such behaviour. They are consistent with the results for one-dimensional networks, showing equally efficient convergence at the larger bandwidths and, if anything, slightly better performance for coarser networks.

3.3.3. Bandwidth dependence and multigrid performance. It will be apparent from the convergence characteristics of algorithm A in Figures 2 and 3 that the breakdown in multigrid performance is not very evident in the first few F-cycles. Indeed, if the average reduction factor for ten F-cycles is plotted against mesh bandwidth for all cases solved, it reveals roughly constant values for bandwidths above about 100 (Figure 5; the filled circles, squares and triangles are results for the different nodal connectivities as indicated). It might thus be assumed that, for the first ten F-cycles, algorithm A is an effective multigrid scheme, albeit a

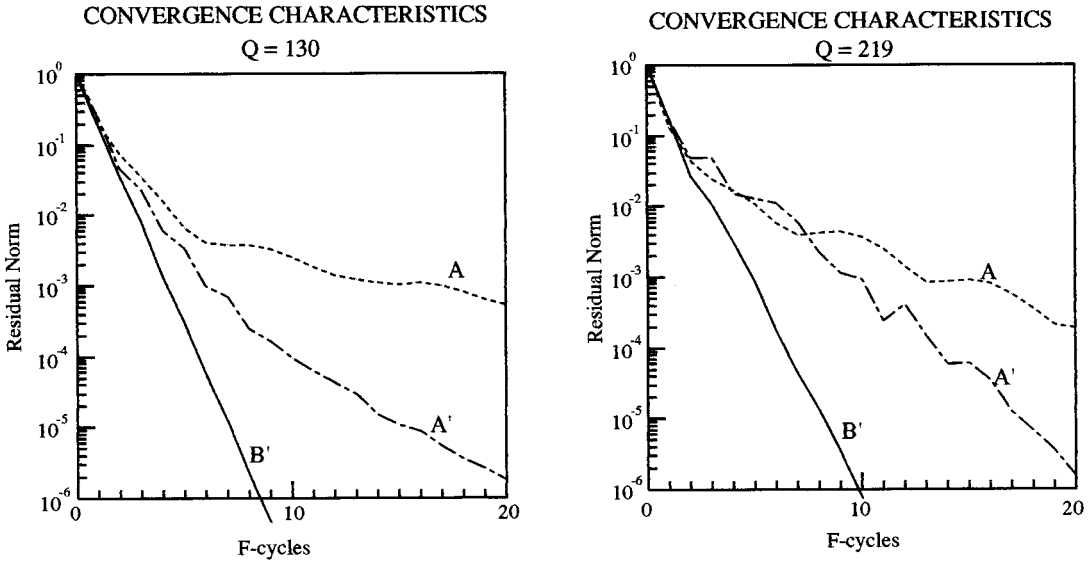

Figure 3. Convergence characteristics for Darcy flow in type-1 (left) and type-2 (right) networks; 17161 nodes (structured): 16784 nodes (unstructured); A (elementary AMG); A′ (elementary AMG with scaled coarse-grid equations); B′(GCR accelerated AMG with scaled coarse-grid equations).

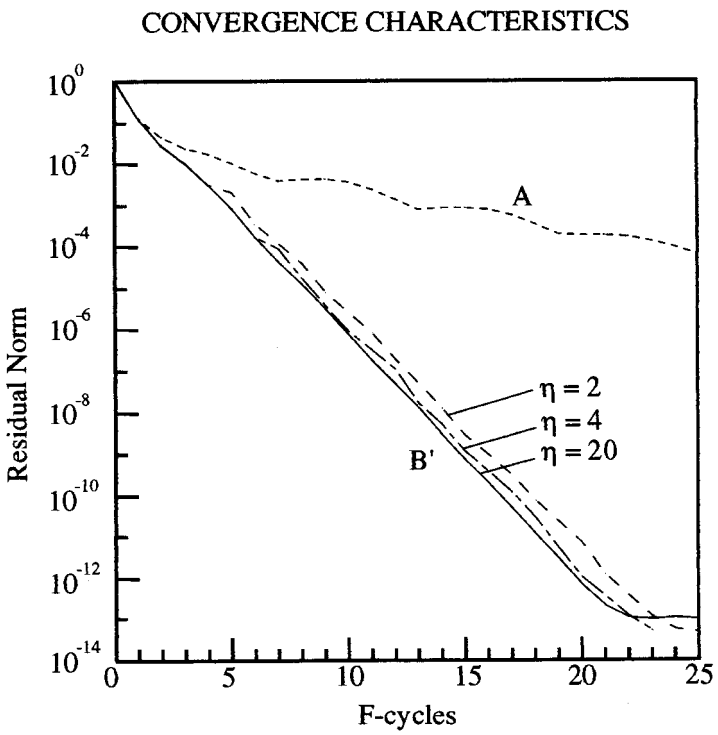

Figure 4. A comparison of convergence characteristics for Darcy flow in unstructured network ($Q = 219$; 16784 nodes); A (elementary AMG) and B′ (GCR accelerated AMG, with scaling of coarse-grid equations).

Table I. Type 1 pipe network

N	900	4225	17 161	84 681
Q	29	64	130	290
$\mu_{\max}$	0.533	0.728	1.34	0.860
$\bar{\mu}$	0.363	0.390	0.486	0.474
F-cycles	13	14	19	18

F(1, 1) residual reduction factors, algorithm A′.

Table II. Type 1 pipe network

N	900	4225	17 161	84 681
Q	29	64	130	290
$\mu_{\max}$	0.240	0.258	0.223	0.250
$\bar{\mu}$	0.201	0.184	0.197	0.228
F-cycles	9	8	9	9

F(1, 1) residual reduction factors algorithm B′.

Table III. Type 2 pipe network

N	1066	4154	16 784	67 305
Q	32	63	131	323
$\mu_{\max}$	0.413	0.479	1.68	2.10
$\bar{\mu}$	0.324	0.327	0.519	0.518
F-cycles	12	12	20	20

F(1, 1) residual reduction factors algorithm A′.

rather inefficient one when compared with algorithm B′ (open circles, squares and triangles). This would be a mistaken assumption. Algorithm B′ corrections are full-bandwidth corrections, in so far as the convergence of Figure 4 is fairly uniform and unlimited within the range of machine accuracy. Corrections for Algorithm A, on the other hand, will have an incomplete spectrum, as is evident in the non-uniform and incomplete convergences of Figure 2 at large bandwidths. For moderate to low bandwidths, $Q \leq 100$, fine-grid smoothing can compensate to some extent, by addressing the poorly represented components of the error spectrum in the coarse-grid approximation more effectively. Thus, in this region, algorithm A convergence improves as bandwidth is reduced; reduction factors fall (Figure 5) and convergence is maintained to low levels of residual error (Figure 2, $Q = 50$). Thus, for coarse networks, $Q \leq 100$, algorithm A can be considered an effective solution scheme.

3.4. Navier–Stokes flow

3.4.1. The equation system. The Navier–Stokes equations describe a more complex system of coupled vector and scalar fields. They are non-linear and must be solved iteratively, e.g. by Picard's method. During each non-linear iteration, a linearised approximation is assembled using the latest iterate and it is this linearised equation set that is to be solved by AMG. The physical complexity of the system is reflected in the system matrix which is asymmetric, involves both first- and second-order differentials and, because it acts on (and couples) several field variables, takes a block-structured form. The relative importance of first- and second-order differentials depends on the nature of the flow as characterised by the Reynolds number, *Re*. Thus for high-*Re* flows where inertial forces dominate, contributions to matrix coefficients from the viscous forces can be relatively small. It might be expected, therefore, that the coarse-grid inconsistencies associated with elementary restriction and prolongation would be

Table IV. Type 2 pipe network

N	1066	4154	16 784	67 305
Q	32	63	131	323
$\mu_{\max}$	0.252	0.271	0.388	0.345
$\bar{\mu}$	0.183	0.213	0.244	0.247
F-cycles	8	9	10	10

F(1, 1) residual reduction factors algorithm B′.

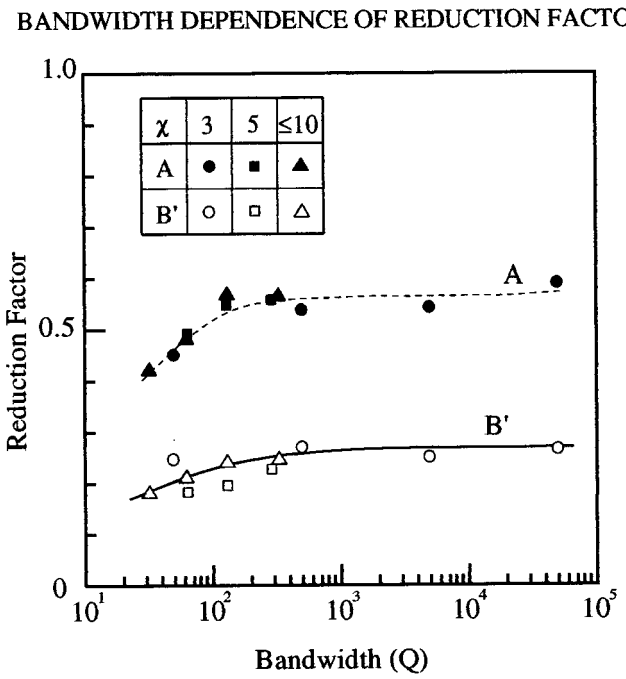

Figure 5. Residual reduction factors (averaged over the first ten F-cycles) versus bandwidth for A (elementary AMG) and B′ (GCR accelerated AMG with approximate scaling of the coarse-grid equations). The nodal connectivities χ, are 3 and 5 for structured networks and between 7 and 10 for unstructured networks.

less troublesome for this type of application, despite its seemingly increased complexity. Whilst this may be the case, it is important to recognise that in the discrete approximation it is the inverse mesh Peclet number, Pe^{-1}, where

$$Pe = ReQ^{-1}, \tag{16}$$

which determines the relative importance of the viscous contributions to the matrix coefficients. Since the mesh bandwidth Q must be sufficiently large to ensure an adequate resolution of the flow, viscous contributions are always likely to be significant, especially for low flow-regions and for regions of high spatial resolution. The performance of an elementary AMG solver would then be compromised by an inconsistent coarsening of the viscous terms.

To explore the effectiveness of coarse-grid scaling and GCR acceleration in dealing with this problem, the discrete Navier–Stokes equations in the formulation introduced in Reference [8] will be considered. They are derived from a finite volume discretisation of a finite element mesh. The mesh will here be an unstructured assembly of linear triangular elements similar to that outlined by the connection network illustrated in Figure 1(c). Nodes are based at the vertices of the elements and the median dual cells constitute the nodal control volumes. If $\mathbf{v}$ represents the set of nodal velocities, $\mathbf{v}_e$ a set of interpolation point velocities within elements and $\mathbf{p}$ the set of nodal pressures, then enforcing the conservation laws for both nodal control volumes and element subcontrol volumes delivers the following set of algebraic equations:

$$\mathbf{A}(\mathbf{v}_e)\mathbf{v} + \mathbf{G}\mathbf{p} = \mathbf{s}, \tag{17}$$

$$\mathbf{A}_e(\mathbf{v}_e)\mathbf{v}_e + \mathbf{F}(\mathbf{v}_e)\mathbf{v} + \mathbf{G}_e\mathbf{p} = \mathbf{s}_e, \tag{18}$$

$$\mathbf{D}\mathbf{v}_e = 0, \tag{19}$$

where $\mathbf{A}$ and $\mathbf{G}$ are here the nodal advection–diffusion and gradient operators respectively; $\mathbf{A}_e$ and $\mathbf{F}$ are each part of the advection–diffusion operator for elements; $\mathbf{G}_e$ is the element gradient operator; $\mathbf{D}$ is the nodal divergence operator, while $\mathbf{s}$ and $\mathbf{s}_e$ represent the momentum source/sink arrays for the nodal control volumes and for element subcontrol volumes respectively (see Reference [10] for further details).

The matrix $\mathbf{A}_e$ is diagonal, so the solution of Equation (18) is trivial, i.e.

$$\mathbf{v}_e = \mathbf{A}_e^{-1}(\mathbf{s}_e - \mathbf{F}\mathbf{v} - \mathbf{G}_e\mathbf{p}). \tag{20}$$

Direct substitution into Equation (19) enables the following subset of coupled equations to be formed for the nodal variables [**vp**]:

$$\begin{bmatrix} \mathbf{A}(\mathbf{v}_e) & \mathbf{G} \\ (\mathbf{D}\mathbf{A}_e^{-1}\mathbf{F}) & (\mathbf{D}\mathbf{A}_e^{-1}\mathbf{G}_e) \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{s} \\ (\mathbf{D}\mathbf{A}_e^{-1}\mathbf{s}_e) \end{bmatrix}. \tag{21}$$

The solution of Equations (20) and (21) is obtained by direct iteration using a predictor–corrector strategy for $\mathbf{v}_e$ and [**vp**], the AMG solver providing the coupled solution of Equation (21) for [**vp**].

For first-order-accurate solutions, Equation (21) is used as it stands, with the diagonal block A assembled using a first-order (upwind) treatment of advection. For second-order-accurate solutions, the same first-order iteration matrix is employed but a compensating defect-correction is added to the right-hand-side. In this way, non-linear iterations converge to second-order-accuracy, whilst the linear solver has the benefit of an iteration matrix that has robust convergence properties.

3.4.2. Global scaling approximation for Navier–Stokes systems. Elementary AMG coarsening of Equation (21) is implemented in a way which preserves the block structure [10]. As the diagonal block operator $\mathbf{A}$ contains the second-order differentials associated with viscous diffusion, inconsistencies will arise in coarsening this diagonal block. These will depend, as discussed above, on the relative size of the diffusive contribution to the coefficients. In keeping with the global scaling approximation, the relative strength of advection to diffusion, Δ, will be taken to be a simple function of a representative, global, mesh Peclet number, $\langle Pe \rangle$,

$$\Delta = \langle Pe \rangle / \{e^{\langle Pe \rangle} - 1\}, \tag{22}$$

where $\langle Pe \rangle$ is an appropriate norm of mesh Peclet numbers for control cells. The scaling factor then becomes

$$\sigma = 1 - \Delta(1 - \beta^{1/d}). \tag{23}$$

Thus, for low mesh Peclet numbers, Equation (23) reduces to Equation (12) as $\Delta \to 1$, whilst for high mesh Peclet numbers $\sigma \to 1$ as $\Delta \to 0$.

Unfortunately, whilst the application of σ to $\mathbf{K}\mathbf{A}\mathbf{K}^T$ alone should improve consistency, it also reduces the absolute magnitude of coefficients in the diagonal block relative to those in the off-diagonal block, $\mathbf{K}\mathbf{G}\mathbf{K}^T$, and this can have a deleterious effect on the smoothing properties of the coarse-grid matrices. A better conditioning is obtained if σ is applied to both blocks.

This also ensures that the numerical coupling gain in the system is conserved. In this regard, note that the diagonal block ($\mathbf{DA}_e^{-1}\mathbf{G}_e$) also contains multiple discrete difference operators, and that when viscous diffusion dominates $\mathbf{A}_e$, the coarsened block, $\mathbf{K}(\mathbf{DA}_e^{-1}\mathbf{G}_e)\mathbf{K}^T$, should scale as $\lambda_c^{-1}\lambda_c^2\lambda_c^{-1} = \lambda^0$, whereas in practice it will scale as $\lambda_c^{-1}\lambda^2\lambda^{-1} = \lambda_c^{-1}\lambda$, so the system hardens with coarsening. The additional scaling for the velocity–pressure coupling block, $\mathbf{KGK}^T$, compensates for this.

3.4.3. Test problems. The Navier–Stokes test problems are based on two-dimensional fluid flow in a square cavity. Three sides of the cavity are solid, no-slip, boundaries; the remaining side has a prescribed uniform tangential velocity that drives a circulation. The above described finite volume discretisation of this problem gives exceptionally accurate solutions, as described in Reference [10]. No further mention of accuracy will be made here, except to say that the calculations are for a second-order-accurate discrete approximation based on the defect correction method. Attention is focused on linear-solver convergence performance.

For any given problem, the linear-solver convergence rates will depend on the particular discrete linear approximation that is being addressed. This will obviously depend on the discretisation (Q/Pe dependent) and on the location of that approximation on the convergence path followed by the non-linear solver (iteration step and initial condition dependent). Reynolds numbers, Re, and mesh resolving powers, Q, have been chosen to provide problems with mesh Peclet numbers, Pe, lying in the three main regions of interest, $Pe \leq 1$, $Pe \sim 1$ and $Pe \geq 1$ ($0.1 < Re < 10^3$; $52 < Q < 166$). All calculations begin from a zero field initial condition (for both flow and pressure). The particular calculations selected for presentation have been chosen, somewhat arbitrarily, from the early stages of the non-linear iteration sequence (low step numbers). As different problems are being addressed, it is important not to read too much into the differences between the problems, but to focus more on the differences in performance of algorithms A, A′ and B′ in each case. However, it would normally be assumed that the larger the Reynolds and Peclet numbers the more difficult the problem; it actually becomes ill posed in the inviscid limit.

3.4.4. Results. The characteristics will be presented for reductions in residual norm to the level of machine accuracy. For any practical Navier–Stokes solver such a tight convergence would be unnecessary and even wasteful; the equation system is, after all, only a temporary linear approximation. The complete convergence is presented here purely as a guide to the quality of the correction spectrum, a uniform, convergence to machine accuracy implying a full-bandwidth, or close to full-bandwidth, correction spectrum.

3.4.4.1. $Pe \ll 1$. Consider first low values of Pe where diffusive contributions to matrix coefficients dominate. This is the case that is closest to the previous application, except that the solver must now provide a coupled solution for vector-flow and scalar-pressure. Figure 6 shows convergence characteristics for algorithms A, A′ and B′ at two levels of spatial resolution, mesh bandwidths $Q = 52.5$ and 166. For the coarser mesh (Figure 6(a)), all algorithms can be considered to be efficient, in so far as convergence rates are fairly uniform down to machine accuracy, but algorithm B′ has the best convergence rate ($\bar{\mu} = 0.0902$, $\bar{\mu} = 0.219$ and $\bar{\mu} = 0.318$ for B′, A′ and A respectively). Algorithm A performs quite well because at such low resolution, the smoother is able to reach any poorly represented modes of the error spectrum. This is consistent with the results for the previous application (Figure 2(a)).

For the finer mesh however (Figure 6(b)), the inconsistencies in the coarse-grid approximation become apparent and the convergence of algorithm A is then relatively poor. Although convergence does not break down completely, it is sluggish by multigrid standards, $\bar{\mu} = 0.763$; about 150 F-cycles would be required to reduce residuals down to machine accuracy. This is consistent with results of Section 3.3 for the single-variable field, and it implies that at higher resolutions the multigrid performance of algorithm A will eventually break down completely. The scaled algorithm A′ is better, $\bar{\mu} = 0.413$, but with GCR acceleration and scaling, algorithm B′ maintains a much more efficient convergent rate, $\bar{\mu} = 0.166$; no significant deterioration in this rate is to be expected at higher resolutions if consistency with the single-variable results is maintained (Figure 5).

3.4.4.2. $Pe \sim 1$. Next, consider mesh Peclet numbers of order 1, where contributions to matrix coefficients from first- and second-order discrete differentials can be of comparable magnitude. The reduction in the relative size of the second-order differential operators might here be expected to improve the performance of algorithm A relative to A′ and B′. Comparing Figure 6(b) and Figure 7(b) shows that for $Q = 166$ ($Pe \leq 2$), this would appear to be the case. The average reduction factor for algorithm A ($\mu = 0.638$) has improved relative to those for algorithm A′ and B′. Admittedly, algorithms A′ and B′ are somewhat more sluggish for this case, which may be partly due to the increase in Reynolds number and partly due to deficiencies in the global scaling approximation for this transition region of mesh Peclet number. Nonetheless, the accelerated algorithm B′ still gives an improved rate of convergence ($\mu = 0.301$) which is also significantly better than that for algorithm A′ ($\mu = 0.479$).

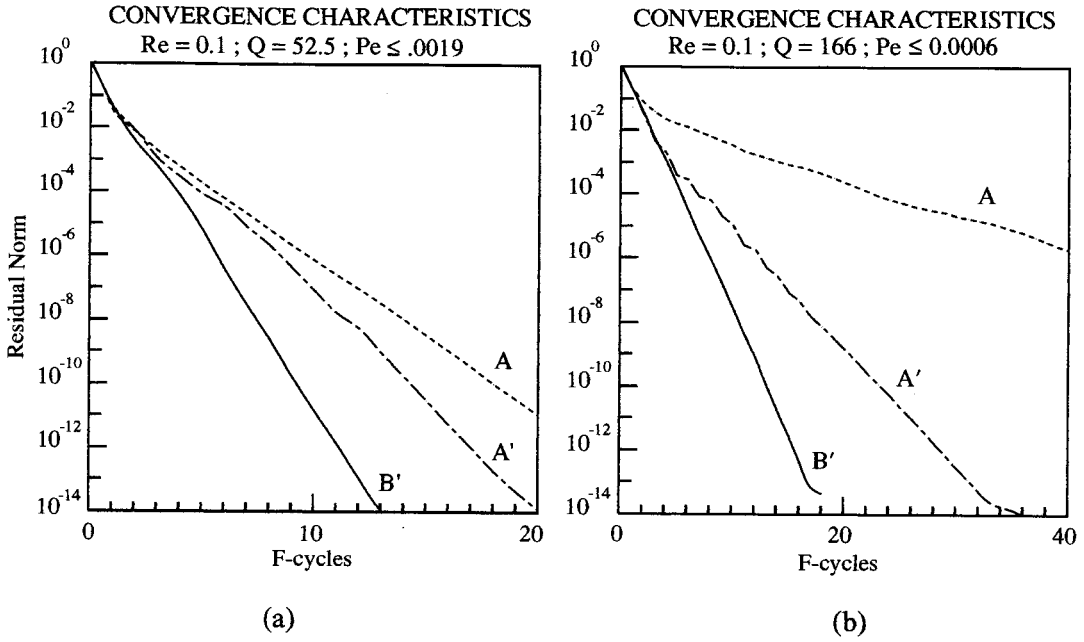

Figure 6. Convergence characteristics for Navier–Stokes flow in a square cavity. (a) $Re = 0.1$, $Q = 52.5$, $Pe \leq 1.9 \times 10^{-3}$, $N = 5340$; (b) $Re = 0.1$, $Q = 166$, $Pe \leq 6.0 \times 10^{-4}$, $N = 59950$.

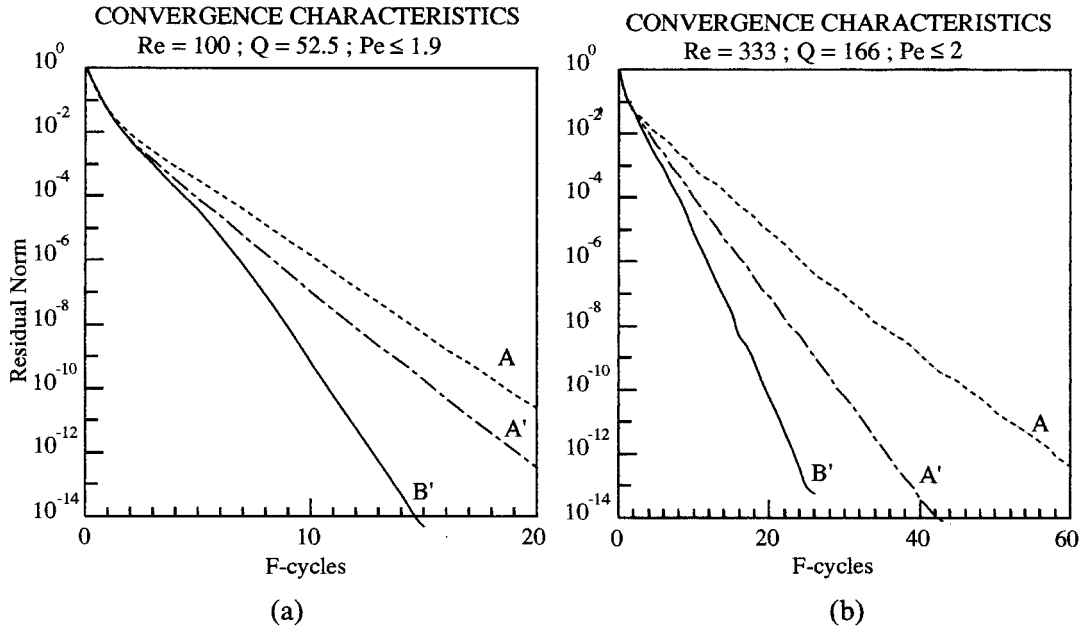

Figure 7. Convergence characteristics for Navier–Stokes flow in a square cavity. (a) $Re = 1.00 \times 10^2$; $Q = 52.5$; $Pe \leq 1.9$; $N = 5340$; (b) $Re = 3.33 \times 10^2$; $Q = 166$: $Pe \leq 2.0$; $N = 59950$.

3.4.4.3. $Pe \gg 1$. Finally, consider Peclet numbers much greater than 1 (Figure 8). Here, at $Pe \sim 19$, scaled and unscaled algorithms are indistinguishable ($\Delta \to 0$ and $\sigma \to 1$), advection dominates diffusion to such an extent that coarse-grid inconsistencies should be greatly reduced. Despite this, convergence rates for algorithm A show no improvement; they are actually worse, $\mu = 0.367$ at $Q = 52.5$; $\mu = 0.710$ at $Q = 166$. This is because the higher Peclet number represents a less elliptic and therefore tougher problem for multigrid and simple point Gauss–Seidel smoothing. Algorithm B′/B, on the other hand, continues to provide very efficient convergence, $\bar{\mu} = 0.102$ at $Q = 52.5$ and $\bar{\mu} = 0.275$ at $Q = 166$, slightly better than the previous lower Reynolds number case. Thus, in this range of application, the GCR control harness enables the solver to address the non-elliptic aspects of the solution more effectively. Algorithm B′ may then be more appropriately described as an AMG preconditioned GCR solver. Note that the convergence rate after the first few F-cycles progressively improves with iteration count. This is typical of well-preconditioned, minimum residual solvers.

Algorithm B′ is thus effective in all three regions of Peclet number, giving significant performance gains in all cases. At low Peclet numbers the GCR control harness forces the solver to address the convergence of those eigenmodes poorly represented in the coarse-grid approximation (GCR accelerated AMG), whilst at high Peclet numbers it forces the solver to better address the essentially non-elliptic aspects problem (AMG preconditioned GCR). The efficiency has been won, however, at the expense of the extra memory required for the GCR accelerator. Therefore, it is important to establish whether the restarted algorithm, GCR(η), can again be used to reduce this storage cost.

3.4.4.4. Restarted GCR acceleration. For symmetric matrix operators, the restarted accelerator GCR(η) was found to be almost as efficient as the full GCR algorithm (e.g. Figure 4), so memory requirements could easily be minimised by limiting η. For asymmetric matrix

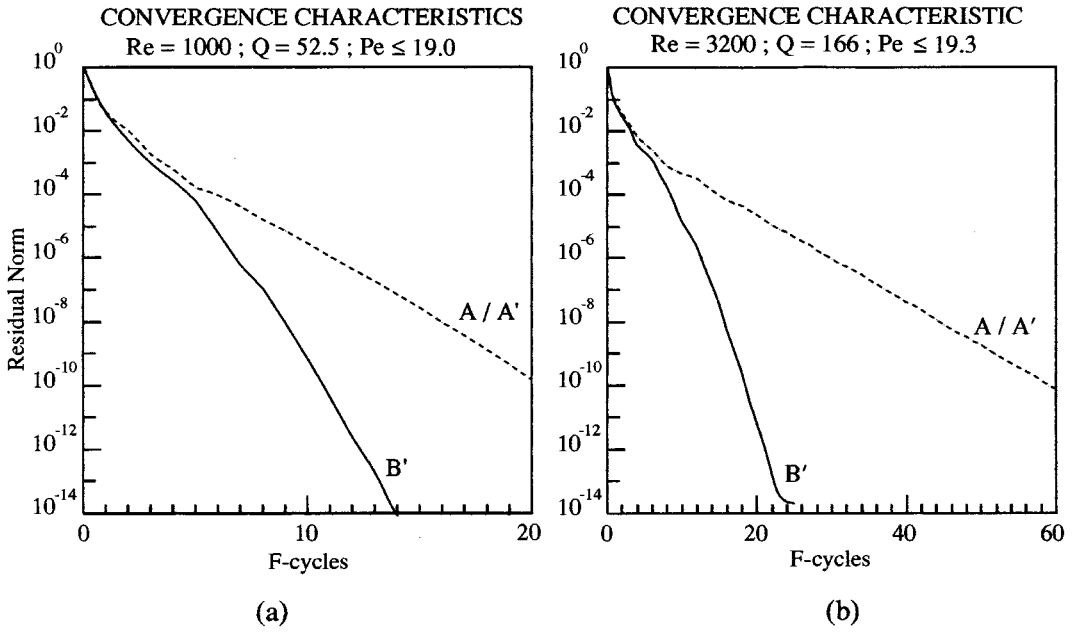

Figure 8. Convergence characteristics for Navier–Stokes flow in a square cavity. (a) $Re = 10^3$; $Q = 52.5$; $Pe \leq 19.0$; $N = 5340$; (b) $Re = 3.2 \times 10^3$; $Q = 166$; $Pe \leq 19.3$; $N = 59950$.

operators, GCR(η) is expected to be less effective, depending on the degree of asymmetry in the advection–diffusion matrix and on the number of steps, η, that available storage capacity will allow. In Table V, reduction factors $\bar{\mu}$ are listed in rows for three driven-cavity problems ($Re = 1$, 333 and 3200) all discretised to a spatial resolving power $Q = 166$, which corresponds to mesh Peclet numbers of 0.006, 2.006 and 19.28 respectively. Values for the accelerated algorithms are listed in columns from left to right in order of decreasing memory requirement. The final column gives values for elementary AMG. The table shows that all the accelerated algorithms give performance gains. The gains are largest (and least sensitive to the value of η) at the lowest value of Pe, i.e. the case having the least asymmetry (consistent with the results for the symmetric operator). For the most asymmetric case, on the other hand, the performance gain is more sensitive to η, but even here the accelerator requiring the least storage space, GCR(2), gives a significant improvement, and the performance difference between GCR(5) and GCR(10) is not large. The restarted accelerator is effective in reducing storage costs without incurring any prohibitive loss in the performance gain.

Table V. F(3, 1) residual reduction factors for Navier–Stokes flow in a square cavity for a mesh resolving power $Q = 166$

Algorithm	B′				A′	A
Accelerator	GCR	GCR(10)	GCR(5)	GCR(2)	None	None
$\bar{\mu}(Pe_{\max} = 0.006)$	0.164	0.176	0.177	0.179	0.413	0.720
$\bar{\mu}(Pe_{\max} = 2.00)$	0.310	0.345	0.360	0.370	0.479	0.638
$\bar{\mu}(Pe_{\max} = 19.3)$	0.275	0.366	0.430	0.480	0.710	0.710

The scaled algorithm without acceleration, A′, requires no extra storage, but the improvements gained are more modest, and are only realised at the low mesh Peclet numbers where scaling is effective.

4. FURTHER COMMENTS ON GENERAL APPLICATION

In the general context of MG solvers, Algorithm B′ may be described as a 'black box' solver for large systems of second-order, discrete-difference equations having positive (at least semi-definite) system matrices. Apart from the equation set, no other problem specific information is required. Its application is not even restricted to problems with an underlying geometrical foundation. For problems with mixed first- and second-order difference operators, however, information on the relative strengths of the operators is also required, but this can be easily accommodated within the framework of the global scaling approximation as a simple adjustment to the global scaling parameter.

The range of mesh bandwidths covered in the one-dimensional tests above (possibly the two-dimensional tests also) exceeds the range of three-dimensional resolution accessible with present computing technology. Therefore, algorithm B′ probably represents an efficient multigrid solver for both existing and foreseeable limits of spatial resolution. Algorithm A is much less effective and, as discussed, is only viable for discretisations of low bandwidth where efficient smoothers are able to address those components of the error spectrum poorly represented in the coarse-grid approximations. For sufficiently high-bandwidth applications the convergence of algorithm A will break down completely.

For Navier–Stokes applications, the efficiency of algorithm B′ does not appear to be sensitive to the *Pe* number; efficient linear-solver convergence has been obtained for discretisations with relatively large values ($Pe > 20$; $Q > 120$). Dispersive truncation errors at high *Pe* numbers are known to degrade the performance of non-linear solvers, to the extent that stagnation and even divergence can occur [10]. Where an apparent convergence is obtained, the solution can be under-diffusive and 'contaminated' by spatial instabilities (wiggles). Therefore, large *Pe* numbers should be avoided and very large values are likely to be impractical. It is for this reason that discretisations with *Pe* numbers greater than 20 have not been presented in the above results, even though efficient linear-solver solutions have been obtained in such cases with algorithm B′.

5. CONCLUSIONS

A simple scaling may be used to reduce inconsistencies in the coarse-grid approximations of AMG solvers based on zero-order intergrid transfer operators. Residual components of the error spectrum that remain poorly represented can then be reduced efficiently by conjugate-gradient acceleration. Thus large sets of second-order, discrete-difference equations, with positive (at least semi-definite) system matrices, may be solved iteratively at convergence rates that are independent of the system bandwidth.

The method represents a simple, robust and cost effective approach to the problem of slowly converging eigenmodes when low-order restriction and prolongation operators are used in AMG algorithms. The alternative, higher order operator approach would require more memory and a larger operation count. It could also be more difficult to implement within the framework of AMG, as algebraic coarsening may not guarantee the grid quality required for a higher order interpolation.

The particular implementation presented here, algorithm B′, uses a simple global scaling approximation together with a GCR accelerator. Alternative accelerators, such as CGS or BICG, may be just as effective and may require less storage. The additional storage required for GCR acceleration can be kept to a minimum by exploiting restart methods.

The algorithm has been tested on an inherently discrete problem (solution of the equations for nodal pressures in a pipe network), and on a continuum field problem in a discrete approximation (solution of the coupled Navier–Stokes equations for fluid velocities and pressures in a driven cavity). It was found to be efficient in both applications.

ACKNOWLEDGMENTS

I am indebted to Ben MacGregor and Guy Robinson for reading and commenting on this paper.

REFERENCES

1. P.W. Hemker, 'On the order of prolongations and restrictions in multigrid procedures', *J. Comput. Appl. Math.*, **32**, 423–429 (1990).
2. A. Brandt, 'Multi-level adaptive techniques (MLAT) for partial differential equations: ideas and software', in J. Rice (ed.), *Proc. Symposium on Mathematical Software*, Academic, New York, 1977, pp. 277–318.
3. W. Hackbush, *Multigrid Methods and Applications*, Springer, Berlin, 1985.
4. J. Ruge and K Stuben, 'Algebraic multigrid', in S. Cormick (ed.), Multigrid Methods, Frontiers in Applied Mathematics, vol 5, *SIAM*, Philadelphia, 1987.
5. M.-H. Lallemand, H. Steve and A. Dervieux, 'Unstructured multigridding by volume agglomerating: current status', *Comput. Fluids*, **21**, 397–433 (1992).
6. B. Koobus, M.-H. Lallemand and A. Dervieux, 'Unstructured volume-agglomeration MG: solution of Poisson equation', *Int. J. Numer. Methods Fluids*, **18**, 27–42 (1994).
7. R.D. Lonsdale, 'An algebraic multigrid solver for the Navier–Stokes equations on unstructured meshes', *Int. J. Numer. Methods Heat Fluid Flow*, **3**, 3–14 (1993).
8. R. Webster, 'An algebraic multigrid solver for Navier–Stokes problems', *Int. J. Numer. Methods Fluids*, **18**, 761–780 (1994).
9. R. Webster, 'An algebraic multigrid solver for Navier–Stokes problems in the discrete second-order approximation', in N.D. Melson, T.A. Manteuffel and S.F. Cormack (eds.), *NASA Conference Publication*, Copper Mountain Conference on Multigrid Methods, April 1995.
10. R. Webster, 'An algebraic multigrid solver for Navier–Stokes problems in the discrete second-order approximation', *Int. J. Numer. Methods Fluids*, **22**, 1103–1123 (1996).
11. R. Kettler, 'Analysis and comparison of relaxation schemes in robust multigrid and conjugate gradient methods', Multigrid Methods, W. Hackbush and U. Trottenberg (eds.), *Lecture Notes in Mathematics 960*, Springer, Berlin, 1982, pp. 502–534.
12. S.C. Eisenstat H.C. Elman and M.H Schultz, 'Variational iterative methods for nonsymmetric systems of linear equations', *SIAM J. Numer. Anal.*, **20**, 345–357 (1983).
13. R. Fletcher, 'Conjugate gradient methods for indefinite systems', *Lecture Notes in Mathematics S06*, Springer-Verlag, Berlin, Heidelberg, New York 1976, pp. 73–89.
14. P. Sonneveld, P. Wesseling and P.M. Elman and P.M. deZeeuw, 'Multigrid and conjugate gradient methods', in D.J. Paddon and H. Holstein (eds.), *Multigrid Methods for Integral and Differential Equations*, 1985, IMA Conference Series, no. 3, Clarendon Press, Oxford, 1985.

