INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
*Int. J. Numer. Meth. Fluids* 2001; **36**: 743–772 (DOI: 10.1002/fld.153)

# Performance of algebraic multi-grid solvers based on unsmoothed and smoothed aggregation schemes

R. Webster*

*Roadside, Harpsdale, Halkirk, Caithness, KW12 6UN, U.K.*

## SUMMARY

A comparison is made of the performance of two algebraic multi-grid (AMG0 and AMG1) solvers for the solution of discrete, coupled, elliptic field problems. In AMG0, the basis functions for each coarse grid/level approximation (CGA) are obtained directly by unsmoothed aggregation, an appropriate scaling being applied to each CGA to improve consistency. In AMG1 they are assembled using a smoothed aggregation with a constrained energy optimization method providing the smoothing. Although more costly, smoothed basis functions provide a better (more consistent) CGA. Thus, AMG1 might be viewed as a benchmark for the assessment of the simpler AMG0. Selected test problems for D'Arcy flow in pipe networks, Fick diffusion, plane strain elasticity and Navier–Stokes flow (in a Stokes approximation) are used in making the comparison. They are discretized on the basis of both structured and unstructured finite element meshes. The range of discrete equation sets covers both symmetric positive definite systems and systems that may be non-symmetric and/or indefinite. Both global and local mesh refinements to at least one order of resolving power are examined. Some of these include anisotropic refinements involving elements of large aspect ratio; in some hydrodynamics cases, the anisotropy is extreme, with aspect ratios exceeding two orders. As expected, AMG1 delivers typical multi-grid convergence rates, which for all practical purposes are independent of mesh bandwidth. AMG0 rates are slower. They may also be more discernibly mesh-dependent. However, for the range of mesh bandwidths examined, the overall cost effectiveness of the two solvers is remarkably similar when a full convergence to machine accuracy is demanded. Thus, the shorter solution times for AMG1 do not necessarily compensate for the extra time required for its costly grid generation. This depends on the severity of the problem and the demanded level of convergence. For problems requiring few iterations, where grid generation costs represent a significant penalty, AMG0 has the advantage. For problems requiring a large investment in iterations, AMG1 has the edge. However, for the toughest problems addressed (vector and coupled vector–scalar fields discretized exclusively using finite elements of extreme aspect ratio) AMG1 is more robust: AMG0 has failed on some of these tests. However, but for this deficiency AMG0 would be the preferred linear approximation solver for Navier–Stokes solution algorithms in view of its much lower grid generation costs. Copyright © 2001 John Wiley & Sons, Ltd.

KEY WORDS: algebraic multi-grid; coupled fields; scaled coarse grid approximations; smoothed aggregation

---

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

*Received May 2000*
*Revised August 2000*

## 1. INTRODUCTION

In a recent paper [1] an algebraic multi-grid (AMG) solver is presented for solving elliptic field problems. The basis functions of the coarse grid approximations (CGAs) are obtained directly by aggregation, so the inter-grid transfer operators have an elementary sparse matrix representation with entries of either zero or unity. They need not therefore be stored explicitly as real arrays; an integer mapping will suffice. They are thus inexpensive, both in storage and in operation count. However, the use of such operators for elliptic problems would normally result in an inconsistent CGA. In Reference [1] the inconsistency is reduced by scaling the coarse level matrices. Sluggish modes of the error spectrum (associated with remaining inconsistencies) have their convergence accelerated using a conjugate residual control harness. This strategy, it was suggested, could be computationally more economic than the alternative approach of attempting a more consistent CGA using better but more costly basis functions.

Improved basis functions can be obtained by smoothed aggregation using an iterative energy optimization method [2]. This minimizes the sum of the energy norms of the basis functions subject to certain constraints. These ensure that the supports for the basis do not extend beyond the nearest neighbours of aggregates. The increased complexity of inter-grid operators is thereby checked.

The unsmoothed scheme, equivalent to zero cycles of optimization and hence designated AMG0, may thus be compared with the $n$-cycle optimized scheme (AMG$n$; $n \geq 1$). The design of AMG$n$ is similar to the simplest of the design schemes proposed in Reference [2] for scalar fields. Convergence of the optimization is guaranteed if system matrices are symmetric positive definite. However, in view of the fact that matrices for the test problems considered here may not have this property, improvement in the CGA beyond that for $n = 1$ cannot be guaranteed. Moreover, where the CGA can be optimized, the improvement in solver performance may not justify the additional investment. Thus, for this particular investigation, just one cycle of optimization is chosen for smoothing, even though better convergence rates can be achieved with $n > 1$ for some cases. Note that AMG1 is equivalent to the Jacobi-smoothed aggregation scheme of Vanáek *et al.* [3].

The increased complexity of the smoothed transfer operators necessitates storage of a broader bandwidth matrix (of non-integer entries), which will obviously incur a higher operation count. On the other hand, it provides an improved CGA, which will permit larger step changes in grid size and hence fewer grids. The higher operator complexity could thus be offset by a lower grid complexity, and hence possibly a lower overall algebraic complexity. Moreover, if convergence factors are improved, the extra time invested in grid generation could be offset by shorter solution times. It is not obvious, therefore, that the simpler AMG0 solver will have the performance advantage anticipated. Any advantage it does have may also be problem-dependent. Performances are therefore compared for a range of applications covering scalar, vector and coupled vector–scalar fields. Each has been discretized using various mesh types covering a wide range of mesh bandwidths.

## 2. SOLUTION METHOD AND ALGORITHMS

### *2.1. Equation system and its iterative solution*

A large system of coupled algebraic equations

$$\mathbf{Au} = \mathbf{b} \tag{1}$$

defines the discretized field problem to be solved, where $\mathbf{u}$ is the field solution (a generalized vector which may represent, in a physical sense, scalar, vector or coupled vector–scalar fields); similarly $\mathbf{b}$ is the field source vector and $\mathbf{A}$ is the matrix that characterizes the discrete field equations. For continuum field applications, this discrete system may represent a finite element or a finite difference approximation. Where more than one field is involved the matrix will be block structured. Diagonal blocks represent the intra-field (spatial) coupling, off-diagonal blocks represent the inter-field coupling. The systems do not have to be of the same size, so there need not be a one-to-one correspondence between the equations for one field and those for another. Thus, different finite elements or different nodal distributions may be used for different fields in the system.

The solution starts from an initial guess, $\mathbf{u}^0$, and an associated residual error, $\mathbf{r}^0$, where

$$\mathbf{r}^0 = \mathbf{b} - \mathbf{Au}^0 \tag{2}$$

A correction, $\mathbf{v}$, is sought, $\mathbf{Av} = \mathbf{A}(\mathbf{u} - \mathbf{u}^0) = \mathbf{r}^0$, within the framework of a convergent iterative scheme so that after $n$ iterations, the correction $\mathbf{v}^n$ is given by

$$\mathbf{Av}^n = \mathbf{r}^{n-1} \tag{3}$$

The approximate solution to these equations, $\mathbf{d}^n \cong \mathbf{v}^n$, is obtained by applying an easily computable approximate inverse, $\Gamma \cong \mathbf{A}^{-1}$, to the residual $\mathbf{r}^{n-1}$

$$\mathbf{d}^n = \Gamma\mathbf{r}^{n-1} \tag{4}$$

The closer $\Gamma$ approximates $\mathbf{A}^{-1}$, the more rapid the convergence is. In particular, if $\Gamma$ is such that the spectrum of $\mathbf{d}^n$ matches that of $\mathbf{r}^{n-1}$, and hence $\mathbf{v}^n$, then successive residuals will be simply proportional and convergence will be characterized by a single reduction factor, $\rho = \|\mathbf{r}^n\|/\|\mathbf{r}^{n-1}\|$ and hence $\mathbf{r}^n = \rho^n\mathbf{r}^0$. To construct a correction with a spectrum that closely approximates $\mathbf{v}^n$, an algebraic multi-grid relaxation method is used.

### *2.2. AMG method and the F-cycle algorithm*

Such a procedure begins by obtaining a narrow-band, high-wavenumber correction, $\mathbf{d}^n$, using a simple local relaxation scheme for $\Gamma$. Corrections at lower wavenumbers are then sought from a lower bandwidth (coarse level) representation, obtained with a suitable algebraic restriction procedure. Thus, if $\mathbf{K}^l$ is the restriction operator at level $l$, applying it to Equation (3) will generate a smaller equation set at a coarser grid level, $l+1$

$$\mathbf{A}^{l+1}\mathbf{v}^{l+1} = \mathbf{r}^{l+1} \tag{5}$$

where

$$\mathbf{A}^{l+1} = \mathbf{K}^l\mathbf{A}^l[\mathbf{K}^l]^T$$

Here the iteration count superscript is dropped (understood) and replaced with the grid level superscript. Thus, if $\mathbf{r}^{l+1}$ is derived on the basis of the current residual error

$$\mathbf{r}^{l+1} = \mathbf{K}^l(\mathbf{r}^l - \mathbf{A}^l\mathbf{d}^l) \tag{6}$$

then a solution of Equation (5) can be used to improve $\mathbf{d}^l$ by increasing its bandwidth

$$\mathbf{d}^l \Leftarrow \mathbf{d}^l + [\mathbf{K}^l]^T\mathbf{v}^{l+1} \tag{7}$$

Clearly, Equation (5) has the same form as Equation (3), so the procedure can be repeated to generate successively smaller equation sets and successively lower wavenumber corrections, until at some level, $G$, an ‘exact’ solution by direct inversion, $\mathbf{v}^G = [\mathbf{A}^G]^{-1}\mathbf{r}^G$, becomes practicable. Thus, with a suitable recursion scheme, a full bandwidth correction can be assembled that will transform the initial $\mathbf{d}^n$ into a good approximation for $\mathbf{v}^n$. A simple recursion, the V-cycle scheme, and also a nested recursion, the full multi-grid V-cycle (FMV-cycle) scheme, feature in the algorithm AMG($l$, $\gamma$, cycle, $v_2$, $v_1$) (Figure 1). Inclusion of an $l > 1$ restriction in the second block gives the simpler F-cycle scheme. A pre-conditioning, $\mathbf{S}^{v_1}$, with $v_1$ ($v_1 \geq 0$) applications of a local relaxation operator $\mathbf{S}$ prior to each restriction (6) is included in the algorithm. Similarly, each prolongation (7) is followed by a post-smoothing, $\mathbf{S}^{v_2}$, representing $v_2$ ($v_2 > 0$) applications of $\mathbf{S}$ to smooth out any high-wavenumber errors generated. Argument $\gamma$ is a nest marker. The operator $\mathbf{F}(v_2, v_1)$ will subsequently be used to represent the application of the FMV/F-cycle options.

```
c = $l$ + 1
if ($l$ < G) then
  $\mathbf{d}^l \Leftarrow \mathbf{S}^{\nu_1}\mathbf{d}^l$
  $\mathbf{r}^c = \mathbf{K}^l(\mathbf{r}^l - \mathbf{A}^l\mathbf{d}^l)$
  call AMG(c,1,cycle,$\nu_2$,$\nu_1$)
else
  $\mathbf{d}^G = [\mathbf{A}^G]^{-1}\mathbf{r}^G$
endif
$\mathbf{d}^l \Leftarrow \mathbf{S}^{\nu_2}(\mathbf{d}^l + [\mathbf{K}^l]^T\mathbf{d}^c)$
if ($l$ <G and $\gamma$ = 1 and cycle /= "V" ) then
   for i = c to G
      $\mathbf{d}^i = 0$ ; $\mathbf{r}^i = 0$
   $\mathbf{d}^l \Leftarrow \mathbf{S}^{\nu_1}\mathbf{d}^l$
   call AMG($l$,0,cycle,$\nu_2$,$\nu_1$)
endif
```

Figure 1. Recursive algorithm *AMG*($\lambda$, $\gamma$, cycle, $v_2$, $v_1$) an implementation of the V- or FMV-cycle schedules.

### *2.3. AMGCR algorithms*

$\mathbf{F}(\nu_2, \nu_1)$ will only produce a full bandwidth correction if the equation sets of the coarse grids are fully consistent representations of the reference fine grid equation set. Anticipating that this may not always be fully realized, the iteration cycles are driven with a minimum residual accelerator. The shorthand GCR for generalized conjugate residual is often used. Here the combination of AMG F-cycle and GCR algorithms will be referred to as the AMGCR algorithm. GCR algorithms can be very effective in accelerating the reduction of those modes of the error spectrum not well represented in the coarse level approximations. One disadvantage, however, is that the amount of storage required increases with the iteration count. To limit the amount actually used a restart strategy is commonly adopted. After say $\eta$ iterations, the algorithm is restarted with the latest solution $\mathbf{u}^{\eta}$ as the initial guess $\mathbf{u}^0$. The number of orthonormalized correction vectors is thereby limited to $\eta$. The notation GCR($\eta$) is commonly used. Introducing the pre-specified tolerance level for convergence, $\varepsilon$, and reintroducing the iteration index superscript, $n$, the algorithm AMGCR($\eta$) is set out in Figure 2. However, depending on whether $\mathbf{F}(\nu_2, \nu_1)$ is based on the scaled or the smoothed-aggregation coarse grid approximations (Section 3), future reference will still be made to AMG0 and AMG1 solution algorithms respectively; it being understood that both solvers are driven with a GCR control harness.

### *2.4. Solution smoothing*

A Gauss–Seidel smoothing operator, $\mathbf{S}$, may be repeatedly applied to the solution wherever the system matrix is positive definite. For coupled-field systems that do not satisfy the $\mathbf{M}$ matrix property, repeated application of $\mathbf{S}$ could be unstable unless some protective damping is employed. A more complex smoothing algorithm is thus required, which involves the calculation and application of an optimum damping for every smoothing sweep. Since damping is an implicit part of generalized minimum residual algorithms, as in the GCR control harness above, a GCR($\eta$) algorithm can also be used to control the repeated application of the Gauss–Seidel smoother, $\mathbf{S} = (\mathbf{L} + \mathbf{D})^{-1}$, where $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U}$ is the usual

```
while ( $\| \mathbf{r}^n \| \geq \varepsilon$ ) do
    $n = 0$ ;  $\mathbf{r}^0 = ( \mathbf{b} - \mathbf{A}\, \mathbf{u}^0 )$
    while ( $n < \eta$ and $\| \mathbf{r}^n \| > \varepsilon$ ) do
        $n = n + 1$
        $\mathbf{d}^n = \mathbf{F}(\nu_2,\nu_1)\ \mathbf{r}^{n-1}$
        $\mathbf{z}^n = \mathbf{A}\, \mathbf{d}^n$
        for ( $i = 1$ , $n-1$ ) do
            $\alpha^{(n,i)} = \langle \mathbf{z}^n , \mathbf{z}^i \rangle / \langle \mathbf{z}^i , \mathbf{z}^i \rangle$
            $\mathbf{z}^n = \mathbf{z}^n - \alpha^{(n,i)} \mathbf{z}^i$
            $\mathbf{d}^n = \mathbf{d}^n - \alpha^{(n,i)} \mathbf{d}^i$
        $\omega = \langle \mathbf{r}^{n-1} , \mathbf{z}^n \rangle / \langle \mathbf{z}^n , \mathbf{z}^n \rangle$
        $\mathbf{u}^n = \mathbf{u}^{n-1} + \omega\, \mathbf{d}^n$
        $\mathbf{r}^n = \mathbf{r}^{n-1} - \omega\, \mathbf{z}^n$
    $\mathbf{u}^0 = \mathbf{u}^n$
```

Figure 2. Algorithm *AMGCR*($\eta$) AMG preconditioned GCR.

splitting into lower-triangular, diagonal and upper-triangular blocks. The smoothing algorithm chosen is thus identical to the above AMGCR($\eta$) algorithm except that the $\mathbf{F}(\nu_2, \nu_1)$ operator is replaced by $(\mathbf{L}+\mathbf{D})^{-1}$, with $\eta=\nu_1$ or $\eta=\nu_2$ for pre- and post-smoothing respectively.

## 3. COARSE LEVEL APPROXIMATIONS

The consistency of the coarse level approximations depends on how appropriate the restriction operator, **K**, and prolongation operator, $\mathbf{K}^T$, are for the equations being approximated. Hemker [4] has shown that if $m_r$ and $m_p$ are the lowest-order polynomials (each $+1$) that are faithfully interpolated, and if $m$ is the order of the partial differential field equation, then the following condition must be satisfied:

$$m_r + m_p > m \tag{8}$$

The simplest restriction operator is an aggregation operator that has the effect of simply adding selected equations together. **K** in that case would be represented by an elementary matrix consisting of zeros and ones (obtained by adding together the appropriate rows of the unit matrix). This would provide a zero-order interpolation and would thus only be suitable for fields described by first-order differentials ($m_r + m_p = 2$).

### *3.1. Coarse level approximation scaling*

It has long been known that the coarse level approximations for second-order partial differential equations obtained using zero-order transfer operators can have their consistency improved by a simple scaling of the coarse level matrix [5,6]. As mentioned above, for the scaled AMG algorithm of Reference [1], a global scaling approximation, $\sigma$, is employed

$$\sigma = \beta^{-1/d} \tag{9}$$

where $\beta$ is the grid reduction factor, the ratio of the sizes of successive levels

$$\beta = n^{l+1}/n^{l} \tag{10}$$

where $n^{l+1}$ and $n^l$ are the numbers of nodal equations (free from Dirichlet constraints) on levels $l+1$ and $l$ respectively, and $d$ is the topological dimension of the system. Thus, for scaled aggregation coarsening, the coarse level matrix $\mathbf{A}^{l+1}$ is given by

$$\mathbf{A}^{l+1} = \sigma \mathbf{K}^{l}\mathbf{A}^{l}[\mathbf{K}^{l}]^{T} \tag{11}$$

### *3.2. Aggregation smoothing*

For the alternative AMG*n* approach, $\mathbf{K}^l$ and $[\mathbf{K}^l]^T$ are improved using the constrained energy optimization scheme. Thus, if $\mathbf{P}_n^l$ is the improved prolongation operator after $n$ optimization cycles then

$$\mathbf{P}_n^l = \mathbf{P}_{n-1}^l - \omega\tilde{Z}(\mathbf{D}^{-1}(\mathbf{N}*\mathbf{A}^l\mathbf{P}_{n-1}^l)); \quad n \geq 1: \ \mathbf{P}_0^l = [\mathbf{K}^l]^T \tag{12}$$

and

$$\mathbf{A}^{l+1} = [\mathbf{P}_n^l]^T\mathbf{A}_l\mathbf{P}_n^l \tag{13}$$

where $\mathbf{D}$ is the diagonal of $\mathbf{A}^l$ and $\omega$ is a damping factor. The matrix $\mathbf{N}$ has entries of 0 or 1 and defines the non-zero structure of $\mathbf{P}_n^l$; $\mathbf{N}\in\mathbf{M}^{n^l \times n^{l+1}}$, the vector space of $n^l \times n^{l+1}$ matrices. So the term-by-term product $(\mathbf{N}*\mathbf{A}^l\mathbf{P}_{n-1}^l)$ ensures that the supports for the smoothing $\mathbf{A}^l\mathbf{P}_{n-1}^l$ do not extend beyond those allowed by $\mathbf{N}$. The orthogonal projection $\tilde{Z}$ in space $N$ $(N = \{\mathbf{P}\in\mathbf{M}\colon \mathbf{N}*\mathbf{P} = \mathbf{P}\})$ to the subspace $Z$ $(Z = \{\mathbf{Q}\in N \mid \Sigma_j^{n^{l+1}} q_{ij} = 0\})$ ensure that entries in $\mathbf{P}_n^l$ satisfy $\Sigma_j^{n^{l+1}} p_{ij} = 1$, as do those in $\mathbf{P}_0^l$ ($[\mathbf{K}^l]^T$ having unit bandwidth and unit entries). However, entries in $\mathbf{P}_n^l$ are allowed to spread to the bandwidth of $\mathbf{N}$ and to take non-integer values, so provision for additional storage of real arrays is required; hence the greater potential cost of AMG*n*. The best choices for $\omega$ and $n$ are problem-dependent. For many applications, AMG1 ($n = 1$; $\omega = 2/3$ Jacobi smoothing) is the most cost effective, even though convergence factors may be better for AMG*n* ($n > 1$; $\omega < 2/3$).

### 3.3. Aggregation

The aggregation defining $\mathbf{K}^l$, $[\mathbf{K}^l]^T$, is a decomposition of the fine level nodes into mutually disjoint subsets, each fine level node contributing to at most one aggregate (hence the unit bandwidth for $[\mathbf{K}^l]^T$). The choice is dictated by two principal requirements, namely, that

1. Aggregated nodes should be part of a strongly coupled neighbourhood.
2. The decomposition should result in coarse level matrices that retain a sparse pattern.

The second of these requirements is easily satisfied for aggregations that will only support the zero-order interpolation of AMG0. More care is required in selecting a decomposition that will support the higher-order interpolation of AMG*n*. The algorithm used for the latter will be described first.

*3.3.1. Aggregation algorithms for smoothed aggregation coarsening*. For any grid level $l$, there will be a set $R$ of $n^l$ nodal equations

$$R = \{1, \ldots, n^l\} \tag{14}$$

Wherever this set is ordered in a particular way, it will be designated $\bar{R}$.

A disjoint decomposition of $R$, $\{C_j^l\}$ $(j = 1, \ldots, n^{l+1})$, is sought by aggregation so that each aggregate, $C_i^l$, gives rise to one node on level $l + 1$. Nodal equations in $R$ are preferentially aggregated according to the strength of coupling between them. If coefficients of $\mathbf{A}^l$ are designated $a_{ij}^l$ $(i = 1, \ldots, n^l; j = 1, \ldots, n^l)$, then following Vanáek *et al.* [2] a strongly coupled neighbourhood, $N_i^l(\varepsilon)$, is defined as

$$N_i^l(\varepsilon) = \{j\colon |a_{ij}^l| \geq (\varepsilon[0.5]^{l-1}[a_{ii}^l a_{jj}^l]^{1/2})\} \cup \{i\} \tag{15}$$

where $0 < \varepsilon < 1$. During the course of the decomposition $|R| = n^{l-k}$ is reduced while $k = |\{C_i^l : i = 1, \ldots, k\}|$ is increased. This is accomplished with the help of a retreating front, $F$, of 'boundary' nodes in $R$, $F \subset R$. Thus, the set $F$ represents the string of $R$ nodes immediately in front of the encroaching $\{C_i^l : i = 1, \ldots, k\}$ domain

$$F = \{ j : |N_j^l(\varepsilon) \cap \{C_i^l : i = 1, \ldots, k\}| > 0\} \tag{16}$$

A preferred subset of $F$, $S$, is defined as

$$S = \{ j : |N_j^l(\varepsilon) \cap \{C_i^l : i = 1, \ldots, k\}| > 1\} \tag{17}$$

the nodes in $F$ that have more than one strong connection to the existing aggregates. As in the case of $R$, wherever the set $S$ is ordered in a particular way it will be designated $\bar{S}$.

The ordering $\bar{R}$ of $R$ is based on coupling strength as follows. If a node $i$, has a most strongly coupled neighbour, $p$, then the strength of the coupling in the neighbourhood, $\mathbf{N}_p^l(\varepsilon)$, is used as the key for ordering node $i$. The ordering of the set $S$ is based on the number $n_S$, of strong connections between a node $j$ and the neighbouring aggregates

$$n_S = |N_j^l(\varepsilon) \cap \{C_i^l : i = 1, \ldots, k\}| \tag{18}$$

The most preferred member of the $\bar{R}$ and $\bar{S}$ sets will be those at the top of the associated storage stacks; these will be designated $\bar{R}(1)$ and $\bar{S}(1)$ respectively.

The decomposition is implemented in three stages (Figure 3). In the first stage, *Aggregation*($\mathbf{A}^l$, $\varepsilon$, Stage 1), a tentative decomposition is established that may not be complete, so a residue of $R$ nodes may remain. The second stage, *Aggregation*($\mathbf{A}^l$, $\varepsilon$, Stage 2), attempts to agglomerate these nodes into existing aggregates but is only permitted to do so if there is a strong connection between the subject node and an aggregate. If free nodes still remain, the third stage, *Aggregation*($\mathbf{A}^l$, $\varepsilon$, Stage 3), completes the decomposition by aggregating these into strongly coupled subsets of one or more nodes.

A graphical illustration, which shows a snapshot of the coarsening process for one grid during the application of *Aggregation*($\mathbf{A}^l$, $\varepsilon$, Stage 1), is presented in Figure 4. This is for an annular domain of otherwise arbitrary shape, which has been discretized on the basis of linear triangular finite elements with nodes at the vertices of the elements. There is a degree of local refinement towards the bottom left of the domain. Coupling strength between nodes is assumed to be related to their proximity, so coarsening begins in the bottom left region and progresses toward the top right. In the illustration it has progressed about half way, with aggregated $C$ nodes distributed behind the front of $F$ nodes. Note that there are trapped pockets of $R$ nodes within the predominantly $C$ domain, which appear singly or in clusters. At an earlier stage, these would have been contained in the $S$ and $F$ subsets, but would have lost this status when they failed to qualify for the $Q$ set, i.e. when no suitable $R$ node neighbour was found that would qualify as a seed for an aggregation. It is to deal with these unresolved nodes that Stages 2 and 3 of the coarsening algorithm is required; it is then that they will either be agglomerated into existing aggregates or be aggregated into clusters (of one or more nodes) to form new $C$ nodes in their own right.

*Aggregation*($\mathbf{A}^l$, $\varepsilon$, Stage1)

```
$\bar{R} \leftarrow R$
for $k = 1$ to $n^l$ do
  $i \leftarrow \bar{R}(k)$
  if ( $i \in R$ ) then
    $S \leftarrow \{i\}$ ; $F \leftarrow \{i\}$
    repeat
      $i \leftarrow \bar{S}(1)$
      $Q = N^l_i(\varepsilon) \cap (R - F)$ ;  $\bar{Q} \leftarrow (Q ;$ ordered $)$
      if ( $Q \neq \emptyset$ ) then
          $j = j + 1$
          $p \leftarrow \bar{Q}(1)$
        $C^l_j \leftarrow N^l_p(\varepsilon)$
          $R \leftarrow R - C^l_j$ ;  $F \leftarrow F - C^l_j$ ;  $S \leftarrow S - C^l_j$
          $F \leftarrow F \cup \{k : k \in R, |N^l_k(\varepsilon) \cap C^l_j| > 0\}$
          $S \leftarrow S \cup \{k : k \in F, |N^l_k(\varepsilon) \cap C^l_j| > 1\}$
      endif
    until $S = \emptyset$
  endif
```

*Aggregation*($\mathbf{A}^l$, $\varepsilon$, Stage2)

```
$\bar{C}^l_k \leftarrow C^l_k$ ( $k = 1,\dots j$ )
for $i = 1$ to $n^l$ do
  if ( $i \in R$ ) then
    for $q = 1$ to $j$ do
      if (( $\bar{C}^l_q \cap N^l_i(\varepsilon)) \neq \emptyset$ ) then
        $C^l_q \leftarrow C^l_q \cup \{i\}$
        $R \leftarrow R - \{i\}$
        exit
      endif
  endif
```

*Aggregation*($\mathbf{A}^l$, $\varepsilon$, Stage3)

```
for $i = 1$ to $n^l$ do
  if ( $i \in R$ ) then
      $j = j + 1$
    $C^l_j \leftarrow N^l_i(\varepsilon)$
    $R \leftarrow R - C^l_j$
  endif
$n^{l+1} = j$
```

Figure 3. Algorithm *Aggregation*($\mathbf{A}^l$, $\varepsilon$, Stages 1–3) for AMG1.

During the course of the decomposition, the covering $\{C^l_j\}$ $(j = 1, \dots, n^{l+1})$ is implicitly stored in the form of a fine-to-coarse mapping, $\mathrm{cf}(1, \dots, n^{l+1})$, where

$$\mathrm{cf}(i) = j; \quad i \in C^l_j \tag{19}$$

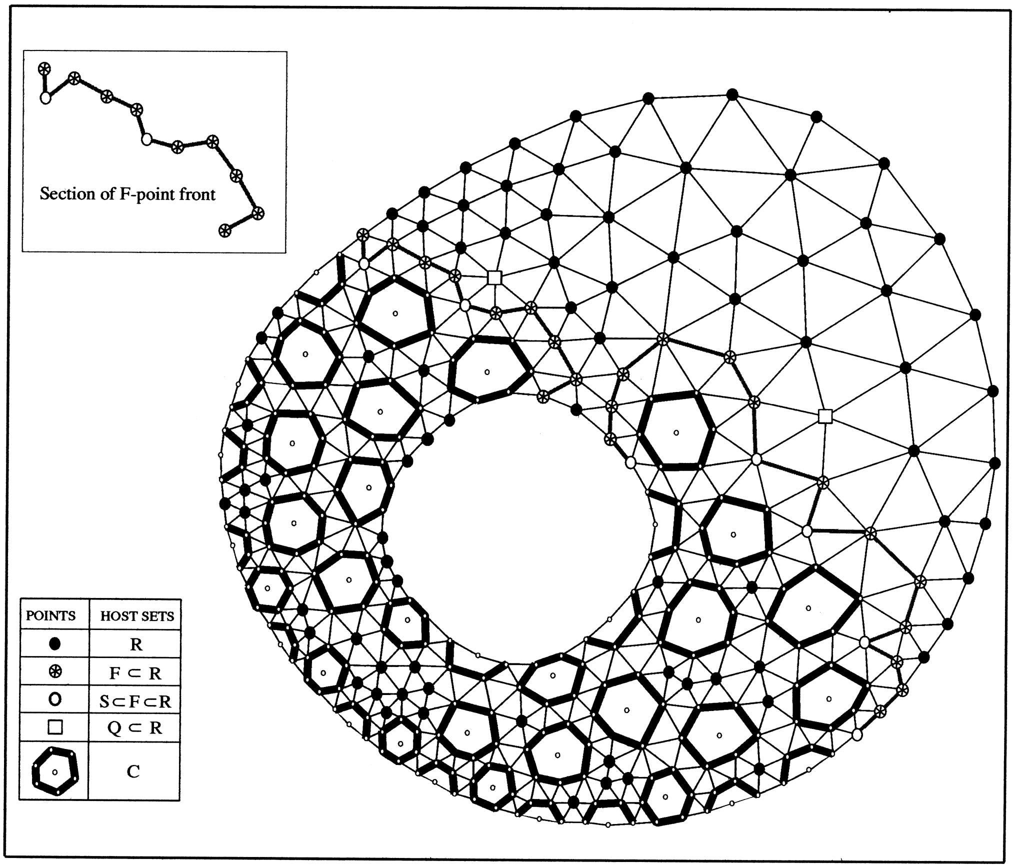

Figure 4. A snapshot of nodal status during coarsening with algorithm *Aggregation*($\mathbf{A}^l$, $\varepsilon$, Stage 1).

The zero-order prolongation, $[\mathbf{K}^l]^T$, is simply related to cf($i$) through

$$([\mathbf{K}^l]^T)_{ij} = \delta_{\mathrm{cf}(i),j} \tag{20}$$

where $\delta$ is the Kronecker delta function. It is smoothed (12) to form the prolongation operator, $\mathbf{P}^l$. Then in turn, $\mathbf{P}^l$ together with $[\mathbf{P}^l]^T$ enables the coarse level system matrix, $\mathbf{A}^{l+1}$ to be constructed (13).

A complete multi-grid hierarchy is obtained by recursion using the algorithm, *Grid_ Generation*($\mathbf{A}^l$, $\varepsilon$), as shown for AMG1 in Figure 5.

*3.3.2. Aggregation algorithms for AMG0.* In principle, the same aggregation scheme could be used for AMG0, the smoothing of basis functions being simply replaced by the scaling of coarse level approximations. However, Stage 1 was based on a retreating front procedure in an

```
$Grid\_Generation( A^l , \varepsilon )$

    $(\mathbf{K}^l)^T \Leftarrow Aggregation(\mathbf{A}^l, \varepsilon$ , Stage1 to Stage3$)$
    $\mathbf{P}^l = ( \mathbf{K}^l)^T - \omega\tilde{Z}(\mathbf{D}^{-1}(\mathbf{N}*\mathbf{A}^l ( \mathbf{K}^l)^T)$
    $\mathbf{A}^{l+1} = [\mathbf{P}^l]^T \mathbf{A}^l \mathbf{P}^l$
    if$(n^{l+1} < n^l)$ $Grid\_Generation( A^{l+1} , \varepsilon )$
```

Figure 5. Recursion algorithm *Grid_Generation*$(\mathbf{A}^l, \varepsilon)$ for AMG1.

attempt to produce the best nodal distribution on the coarse level for first-order linear interpolation. The scaled coarse level approximations, based on lower-order transfer operators, may require a much more gradual coarsening, which the retreating front scheme is unsuited. Thus, for AMG0 additional control of the coarsening is built into the algorithm through the introduction of a target grid reduction factor, $\beta$, and of two more restrictive types of strongly coupled neighbourhoods. The first represents, at most, a strongly coupled pair and will be written without arguments as $N_i^l$. It consists of subject node, $i$, and, if one exists, a strongly coupled neighbour, $p$, which has a strong connection itself. The second type is written $N_p^l(\beta)$, with $\beta$ as an argument. It only exists if partner $p$ exists, $\{p\} = N_i^l - \{i\} \neq \emptyset$, and consists of those neighbours of node $p$, including subject node $i$, that have a coupling strength $|a_{pj}|$, which satisfies

$$|a_{pj}| > 0.2|a_{pi}| \tag{21}$$

subject to the constraint

$$|N_i^l(\beta)| \leq \beta^{-1} \tag{22}$$

i.e. the number in the subset is limited to the nearest integer of $\beta^{-1}$ of the most strongly coupled neighbours. The decomposition is again implemented in three stages (Figure 6), each of which has a similar function to that for the previous case. Argument $\varepsilon$ is absent since the strong coupling parameter has effectively been 'hard wired' into the algorithms through inequality (21).

The recursion algorithm for generating the grid hierarchy, *Grid_Generation*$(\mathbf{A}^l, \beta)$, is even simpler than the previous case since the scaling may be implemented directly as part of the coarse matrix assembly (Figure 7).

## 4. BASIS OF THE INVESTIGATION

This section describes the practical applications selected for the numerical investigations and the reasoning behind those selections. The actual problem definition for the chosen applications follows; this includes the metrics used to quantify the formulation, the approximations adopted and the computational meshes employed. Numerical aspects of solver application are

*Aggregation*( $A^l$, β , Stage1)

```
$\bar{R} \leftarrow R$
for k = 1 to $n^l$ do
  $i \leftarrow \bar{R}(k)$
  if ( $i \in R$ ) then
    $\{p\} = N^l_i - \{i\}$
    if ( $\{p\} \neq \emptyset$ ) then
      $j = j + 1$
      $C^l_j \leftarrow N^l_p(\beta)$
      $R \leftarrow R - C^l_j$
    endif
  endif
```

*Aggregation*( $A^l$, β , Stage2)

```
$\bar{C}^l_k \leftarrow C^l_k$ ( k = 1,...j )
for i = 1 to $n^l$ do
  if ( $i \in R$ ) then
    for q = 1 to j do
        if (( $\bar{C}^l_q \cap N^l_i$ ) $\neq \emptyset$ ) then
          $C^l_q \leftarrow C^l_q \cup \{i\}$
          $R \leftarrow R - \{i\}$
          exit
        endif
  endif
```

*Aggregation*( $A^l$, β , Stage3)

```
for i = 1 to $n^l$ do
  if ( $i \in R$ ) then
    $j = j + 1$
    $C^l_j \leftarrow \{i\}$
    $\{p\} = N^l_i - \{i\}$
    if ( $\{p\} \neq \emptyset$ ) $C^l_j \leftarrow N^l_p(\beta)$
    $R \leftarrow R - C^l_j$
  endif
$n^{l+1} = j$
```

Figure 6. Algorithm *Aggregation*($\mathbf{A}^l$, $\beta$, Stages 1–3) for AMG0.

then described, such as the values used for those free parameters common to both solvers. Finally, solver performance metrics are defined.

### 4.1. Physical applications

The test problems are primarily classical field problems relevant to the physical sciences. However, since these are governed either by discrete difference equations (DDEs) or by partial

```
Grid_Generation( $A^l$, $\beta$ )

    $(\mathbf{K}^l)^T \Leftarrow Aggregation(\mathbf{A}^l, \beta$, Stage1 to Stage3$)$
    $\mathbf{A}^{l+1} = \sigma\,[\mathbf{K}^l]^T \mathbf{A}^l \mathbf{K}^l$
    if($n^{l+1} < n^l$) Grid_Generation( $A^{l+1}$, $\beta$ )
```

Figure 7. Recursion algorithm *Grid_Generation*$(\mathbf{A}^l, \beta)$ for AMG0.

differential equations (PDEs), they will be relevant to any application that can be cast in this form. They cover scalar fields, vector fields and coupled scalar–vector fields, so they should be representative of a very wide range of potential applications. The particular problems addressed here are especially relevant to the engineering sciences and in particular to thermo-, elasto-, electro-, aero-, hydrodynamics branches. They are

1. D'Arcy flow in a pipe network.
2. Fick diffusion in a square cavity.
3. Plane strain elasticity in a material of square cross-section.
4. Stokes/Navier–Stokes flow in (a) square cavity and (b) rectangular duct.

*4.1.1. D'Arcy flow and Fick diffusion.* D'Arcy flow in a pipe network and Fick diffusion in a square cavity have been selected as representatives of scalar field problems. They could have equally well been the flow of electrical current in a resistive network and the diffusion of heat in square solid, the formulations would have been the same. The two problems differ in so far as the first is inherently discrete, the determination of nodal pressures, $\phi_i$, for a pipe network with nodal sources, $s_i$

$$a_{ij}\phi_j = s_i \tag{23}$$

where the system matrix, $a_{ij}$, characterizes the network, while the second is a continuum problem

$$\nabla\cdot[\mathbf{D}\nabla\phi] = s \tag{24}$$

the determination of material concentration, $\phi$, in a continuum with second rank diffusivity tensor, **D**, and a volumetric source $s$. The latter only takes the discrete form of Equation (23) in a finite element/finite difference approximation. While in both cases $a_{ij}$ will be a discrete Poisson-type operator, which for the first case will be a so-called $M$-matrix, symmetric, positive definite when at least one equation is subject to a Dirichlet constraint. For the diffusion case, $a_{ij}$, may not be an $M$-matrix; it depends on details of the discretization and in particular the shape of the finite elements used.

*4.1.2. Plane strain elasticity.* The third case is representative of a vector (tensor) field to solve for a two-dimensional strain field for plane strain elasticity problems. For a body in static equilibrium, all the acting forces must balance. Thus, balancing the surface forces against the body forces gives the static stress equation

$$\nabla \cdot \sigma + \rho \mathbf{g} = \mathbf{f} \tag{25}$$

where $\sigma$ is the stress tensor, $\mathbf{g}$ is the acceleration due to gravity, $\rho$ is the material density and $\mathbf{f}$ is any external force acting. The stress is related to the strain tensor, $\varepsilon_{kl}$, by the equations

$$\sigma_{ij} = c_{ij\,kl}\varepsilon_{kl} \tag{26}$$

where $c_{ij\,kl}$ is a fourth rank stiffness tensor, and the strain is in turn related to the local displacements, $u_k$, $u_l$, according to

$$\varepsilon_{kl} = \frac{1}{2}\left(\frac{\partial u_k}{\partial x_l} + \frac{\partial u_l}{\partial x_k}\right) \tag{27}$$

For an isotropic material most of the coefficient entries of the stiffness tensor reduce to zero and those remaining non-zero can be expressed in terms of Young's modulus, $E$, the rigidity modulus, $G$, and Poisson's ratio, $v$. Expressing the stress tensor in terms of local displacements using the above equations gives the equations for a two-dimensional strain field

$$\frac{E}{(1-v)^2}\left(\frac{\partial^2 u_1}{\partial x_1^2} + \frac{1-v}{2}\frac{\partial^2 u_1}{\partial x_2^2} + \frac{1+v}{2}\frac{\partial^2 u_2}{\partial x_1\,\partial x_2}\right) + \rho g_1 = f_1 \tag{28}$$

$$\frac{E}{(1-v)^2}\left(\frac{\partial^2 u_2}{\partial x_2^2} + \frac{1-v}{2}\frac{\partial^2 u_2}{\partial x_1^2} + \frac{1+v}{2}\frac{\partial^2 u_1}{\partial x_2\,\partial x_1}\right) + \rho g_2 = f_2 \tag{29}$$

where the following relationship for isotropic materials has been used

$$G = E/\{2(1+v)\} \tag{30}$$

Note that in this case there are inter-field couplings so the system matrix will have a block structure with entries in the off-diagonal blocks. For a typical Poisson's ratio of 0.3, the entries can be large and of the 'wrong' sign. The system matrix is therefore unlikely to be an $M$-matrix regardless of the discrete approximation used. The solver smoother efficiency could be adversely affected by these inter-field couplings.

*4.1.3. Stokes/Navier–Stokes hydrodynamics.* The fourth case represents a coupled scalar–vector field test problem, the solution for fluid pressure and flow as governed by the Navier–Stokes equations (or the Stokes equations when the advection of fluid momentum on the flow is not significant in determining the flow field). They are derived from the fundamental conservation laws for momentum and mass. The dynamic equivalent of Equation (25) for an incompressible fluid, Newtons second law of motion, derived from the conservation of momentum is

$$\rho\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u} - \sigma) + \rho \mathbf{g} = \mathbf{f} \tag{31}$$

where field variable $\mathbf{u}$ represents the rate of fluid displacement (velocity). The additional dynamic terms are, firstly, an allowance for the acceleration/deceleration of the fluid and, secondly, an additional contribution to the stresses due to the transport of momentum, $\rho\mathbf{u}$, by $\mathbf{u}$. Note that the incompressibility condition, $\nabla\cdot\mathbf{u}$, has been exploited in casting the advective term in this conservative form. The stress tensor, $\sigma$, has contributions from both fluid pressure, $p$, and the viscous stress tensor, $\tau$, according to

$$\sigma_{ij} = -p\delta_{ij} + \tau_{ij} \tag{32}$$

where for an isotropic incompressible Newtonian fluid of viscosity, $\mu$, the viscous stress, $\tau_{ij}$, is related to the rate of strain tensor, $\dot{\varepsilon}_{ij}$, by

$$\tau_{ij} = 2\mu\dot{\varepsilon}_{ij} \tag{33}$$

Substituting for the stress tensor in Equation (32) gives

$$\rho\frac{\partial\mathbf{u}}{\partial t} + \nabla\cdot[\rho\mathbf{u}\mathbf{u} - \tau] + \nabla p + \rho\mathbf{g} = \mathbf{f} \tag{34}$$

which together with

$$\nabla\cdot\mathbf{u} = 0 \tag{35}$$

the incompressibility condition, constitute the Navier–Stokes equations for $\mathbf{u}$ and $p$. Compared with the previous case, these field equations involve an additional field variable, the scalar pressure, $p$. They are also non-linear in view of the advection term.

### *4.2. Computational domains, meshes and mesh metrics*

For both networks and continua the discrete formulation is based on a division of the calculational domain into an assembly of discrete elements (a finite element mesh). In the case of networks, elements have a single topological dimension (e.g. connections/pipes/struts), regardless of the dimensionality of the domain; they intersect at points. In the case of continua, elements are cells with the same dimensionality as the domain; in this case, neighbouring elements may share common faces, edges and vertex points of intersection. Associated with each element are nodal points at which the discrete field variables are defined. Here they are placed at element vertices, which will permit a linear interpolation. The actual interpolations used will be discussed further below.

A nodal connectivity, $\chi$, is defined as the number of nodes in a cluster (a nodal cluster being a subject node and all its directly connected neighbours). An aspect ratio, $\alpha$, is defined as the maximum of the ratios of the largest to smallest lengths of adjacent connections in a cluster. A mesh or network bandwidth, $Q$, may be defined as a linear mesh size, or the maximum number of element edges (direct nodal connections) along a single continuous line spanning the computational domain.

The mesh bandwidth needed for any problem depends on the complexity of the field solution and the resolution required. A minimum requirement might be that which is necessary for a realistic representation of the boundary geometry. In this investigation, interest is focussed on solver performance rather than solution accuracy, so such considerations are largely ignored. Ideally, multi-grid solver performance should be mesh-independent, i.e. independent of $Q$, $\alpha$ and $\chi$. Performance for a wide range of meshes is therefore investigated. However, in order to cover as wide a range of mesh bandwidth as is possible, very simple rectangular boundary geometries have been selected to ensure that they do not compromise the lower limit of accessible bandwidth. The focus tends to be, therefore, on the complex meshing of a simple domain rather than on the more usual quest for a simple meshing of a complex domain.

The meshes investigated involve constant $\chi$ (structured) and variable $\chi$ (unstructured) mesh types. Token illustrations are given in Figure 8. Actual mean resolutions range over at least one order ($16 < \bar{Q} < 160$) and, where present, anisotropies can be more extreme ($16 < \alpha < 256$). Local refinements in resolution span at least one order. For the unstructured meshes, only triangulations are considered, i.e. the simplest possible two-dimensional elements. Note that mesh type $G$ can contain 'bad elements', i.e. elements with an internal angle exceeding 90°. The most extreme anisotropies are for fluid flow applications and the resolution of boundary layers. Note that where $Q$ is varied by '$n$' orders, problem size could vary by $2n$ or by $3n$

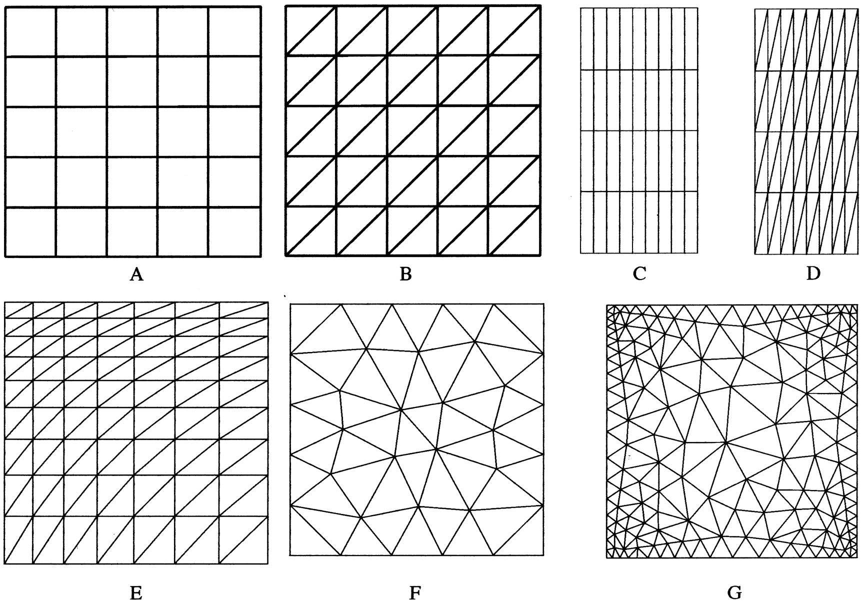

Figure 8. Token meshes: examples of structured homogeneous isotropic meshes (A, B), structured homogeneous anisotropic meshes (C, D), structured inhomogeneous anisotropic mesh (E), unstreuctured quasi-uniform mesh (F) and unstructured non-uniform mesh with local refinements (G).

orders for two- and three-dimensional problems respectively. Similarly, where aspect ratios vary over '*n*' orders, coefficient entries for second-order discrete differences could vary by 2*n* orders.

### *4.3. Discrete formulations*

The discrete formulations are obtained using the 'finite volume' method. Control cells are constructed by dividing each element along the median(s) into equal segments (Figure 9). Each segment intersects one vertex node and thereby contributes to the 'median dual' control cell for that vertex, i.e. the sum of all the segments intersecting a node defines the median dual cell for that node. Thus, the median dividers will also define the control surfaces of the cells (bold lines in Figure 9). Using special interpolation functions, the fluxes of the relevant physical entities at control surfaces can be expressed in terms of the nodal unknowns at the vertices of an element. A simultaneous enforcement of the physical conservation laws for all control cells then generates the required system of coupled algebraic equations, which for continuum field problems represents the discrete approximation for the partial differential field equations (23)–(35).

The resolving power of a discretization will be closely related to the mesh bandwidth $Q$. Here $Q$ is used interchangeably for both mesh bandwidth and resolving power, which is reasonable if the fields involved are resolved to a similar degree. This depends on the interpolation schemes used, which will be discussed further below.

For single fields, the interpolation is straight forward. A simple linear interpolation suffices to give second-order accuracy for Poisson-type operators (third-order truncation error). For coupled scalar–vector fields of fluid dynamics, both first- and second-order differential operators are involved and there is both an intra- and an inter-field coupling. To take account of these factors and to ensure a proper inter-field coupling at element level, a physically based interpolation is adopted [8]. A sub-control cell is constructed within an element. Enforcement of the conservation laws for that sub-control cell gives an expression for the local field

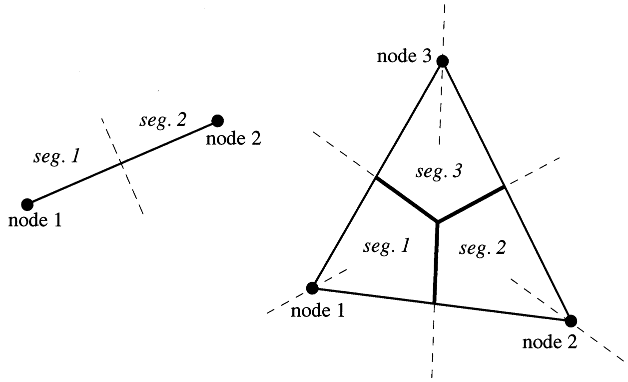

Figure 9. Linear elements used in discretizations.

variables at a single interpolation point (the element centroid) in terms of the nodal field variables. The interpolation point vector can then be used to determine an element flux, which in turn can be used in the enforcement of the conservation laws for the nodal control cells.

### *4.4. Performance metrics*

To quantify, assess and compare the performance of the AMG solvers, it is necessary to choose performance metrics. Both phases of AMG calculations discussed in Section 2, 'grid generation' and 'iterative solution', need to be considered in assessing the overall performance.

*4.4.1. Metrics for grid generation.* It has already been mentioned that the cost of grid generation can be a crucial factor in determining the overall cost effectiveness of AMG$n$, $n > 0$. In particular, grid complexity and algebraic complexity were identified as important factors. Grid complexity, $C_g$, is defined as the ratio of the total number of equations in the multi-level system to the number of equations at the finest level of resolution. Algebraic complexity, $C_A$, is defined as the ratio of the total number of coefficient entries in system matrices for all levels to the number of entries in the system matrix for the finest level. Ideally, $C_A$ and $C_g$ should be independent of mesh bandwidth, $Q$. In the case of grid complexity, this will be realised when the achieved grid reduction factor, $\beta' \geq \beta$, is constant

$$C_g = 1./(1. - \beta') \tag{36}$$

This will be ensured if the target reduction factor is always achieved, $\beta' = \beta$. For $C_A$ to be independent of $Q$ the system matrix bandwidth, $\omega \leq \chi$, also needs to be conserved during coarsening, in which case

$$C_A = \omega/(1. - \beta') \tag{37}$$

*4.4.2. Metrics for iterative solution.* Residual error norms plotted logarithmically against the iteration count provide a convergence characteristic. In this paper, these tended to follow uniformly straight lines, with just small random variations in reduction from cycle to cycle. Examples for AMG0 may be found in Reference [1, pp. 325–327, figures 2–4, plots B′]. Those for AMG1 are similar but generally steeper. As they are otherwise relatively uninteresting, they will not be displayed here. Instead, the average reduction factor, $\bar{\rho}$, will be used as the metric for comparing performance. Thus, for a sequence of $n$ $F$-cycles, $\bar{\rho}$ is defined as

$$\bar{\rho} = \{\|\mathbf{r}^{(n)}\|_2/\|\mathbf{r}^{(0)}\|_2\}^{1/n} = \{\Pi_1^n \rho^{(i)}\}^{1/n} \tag{38}$$

where $\rho^{(i)}$ is the reduction factor for the $i$th iteration, given by

$$\rho^{(i)} = \|\mathbf{r}^{(i)}\|_2/\|\mathbf{r}^{(i-1)}\|_2 \tag{39}$$

$\mathbf{r}^{(i)}$ being the residual following the $i$th iteration cycle.

*4.4.3. Metric for overall performance, work unit*. Overall multi-grid performance is measured in terms of the total amount of work, $W$, required to achieve a pre-specified level of convergence, namely a 12-order reduction in the Euclidean norm of the residual error. This includes the cost of generating the coarse level approximations. For any particular application, it is measured in terms of a reference work unit, $w$, which is usually the computational cost of one sweep of smoothing on the reference equation set (the fine grid) or, alternatively, as the cost of evaluating the residual on the fine grid. Here, the total cost of one application of the smoother is used, which will involve a number ($v_1$, $v_2$) of GCR controlled Gauss–Seidel smoothing sweeps. The unit thus incorporates the overhead costs of the GCR controller. Note that the unit will be $Q^2$-dependent, so that in principle, $W$ should be $Q$-independent for an ideal multi-grid solver. However, $w$ and $W$ are estimated from computing times (CPU seconds). While the use of computing time may be a reliable measure of work done by a machine to complete a task, machine efficiency may vary from one part of a calculation to another and it may also depend on problem size and the available memory. Some caution is, therefore, required when using $W$ as a performance measure. One cannot read too much into small differences between the results for different problem sizes. However, on the assumption that machine overheads will be roughly similar for the two solvers when they address the same problem, $W$ will be used as a comparative performance measure for identical problems. It will not be used to assess the scaling of performance with mesh bandwidth $Q$. The scaling of $C_g$, $C_A$, $n$ and $\bar{\rho}$ will be used as a guide to the overall scaling.

*4.4.4. Quality of multi-grid performance*. As discussed in Section 1, an ideal multi-grid solver is one that reduces all components of the residual error spectrum with equal efficiency, and as such will provide a $Q$-independent convergence. Here it will be similarly defined, but it will also be required that both $C_g$ and $C_A$ be $Q$-independent too. $C_g$, $C_A$ and $\rho$ all need to be $Q$-independent if overall performance $W$ (for a perfect machine) is to be $Q$-independent. In practice, both solvers and machines will fall short of the ideal. The quality is therefore assessed in terms of how close the actual scalings of $C_g$, $C_A$, $n$ and $\bar{\rho}$ compare with the ideal.

*4.4.5. Smoothing performance*. A sufficient amount of work is invested in smoothing to make the solvers effective. Smoothing performance will therefore be reflected in the number of post smoothing sweeps, $v_2$, used.

### *4.5. Choice of solver parameters*

*4.5.1. Convergence*. As indicated above convergence is defined as a 12-order reduction in the Euclidean norm of the residual error, i.e. at that iteration, $n$, when $\|\mathbf{r}^{(n)}\|_2/\|\mathbf{r}^{(0)}\|_2 < 10^{-12}$ is satisfied. This may seem an excessively stringent criterion, since in practice it would be wasteful to demand a reduction of the residual norm to levels much lower than the global error associated with the truncation of the discretization ($\sim Q^{-2}$ for the test problems addressed). It has been imposed purely as a quality check on the correction spectrum; only full bandwidth corrections providing constant convergence rates down to machine accuracy.

*4.5.2. Grid reduction factors*. Grid reduction rates for the smoothed aggregation solver are determined automatically by the coarsening algorithm *Grid_Generation*($\mathbf{A}^l$, $\varepsilon$). More gradual rates of reduction can be imposed, but this has not been tried in these calculations.

For the scaled aggregation solver, the rate of coarsening is controlled by the factor $\beta \leq 0.5$, in algorithm *Grid_Generation*$(\mathbf{A}^l, \beta)$. The lower the value of $\beta$, the lower the complexity of the coarse level approximations and the smaller the memory requirement. This reduced demand for memory however may incur a performance penalty (see below). Calculations have therefore been performed for both large ($\beta = 0.5$) and small ($\beta = 0.14$) values.

*4.5.3. Strong coupling parameter*. The strong coupling parameter, $\varepsilon$, defines a strongly coupled neighbourhood (15). For reasonably uniform isotropic meshes, the performance is not sensitive to the actual value used, providing it is not too large ($\varepsilon < 0.1$). Values larger than about 0.1 can result in inefficient coarsening, high grid complexities and degraded performance. For non-uniform, anisotropic meshes, performance is also insensitive to the precise value used providing it satisfies $\varepsilon_{\text{crit}} \ll \varepsilon < 0.1$, where $\varepsilon_{\text{crit}}$ is the value required to ensure preferential coarsening in strongly connected regions/directions on the fine grid. Values below $\varepsilon_{\text{crit}}$ would result in a 'full coarsening' similar to geometric multi-grid. This would result in poor smoothing and relatively sluggish convergence with the simple local relaxation methods used.

*4.5.4. Optimization parameters*. As mentioned in Section 3.2, a choice of $n = 1$, $\omega = 2/3$ (AMG1) is the most cost effective for many applications, even though better convergence factors may be realised with AMG*n* ($n > 1$, $\omega < 2/3$). In view of this, and in view of the fact that convergence of the optimization scheme cannot be guaranteed for some of the systems considered here, the investigation presented is restricted to a comparison of AMG1 results with those of AMG0. A more thorough examination of AMG*n* performance will be reported elsewhere, both with respect to cost effectiveness and with respect to the robustness of the solution process.

## 5. NUMERICAL RESULTS

### *5.1. Scalar field systems*

*5.1.1. M-matrix systems; D'Arcy flow in pipe networks*. This test is the calculation of laminar pressure-driven flow in pipe networks. Pipe elements 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 (Equation (23)) 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. 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 $10^3$.

Four network types have been selected (cf. Figure 8), structured homogeneous isotropic networks A; unstructured quasi-uniform networks F; structured homogeneous anisotropic ($\alpha = 40$) networks C; unstructured non-uniform networks G, which involve local refinements (of order 10:1). The system matrices are all positive definite but smoothing efficiency may vary significantly. Conceding the possibility that a greater investment in smoothing may be necessary for the highly distorted networks, all calculations have been performed with a fixed allowance of $\nu_2 = 3$, $\nu_1 = 0$ post- and pre-smoothing sweeps respectively; the post-smoothing being controlled by GCR. The results for a one-order range of network $Q$ (two-orders range of problem size) are presented in Tables I and II for AMG1 ($\varepsilon \leq 0.06$) and AMG0 ($\beta = 0.14$) solvers respectively. In these (and all subsequent table sets) the total amount of work done, $W_T$ (work units), is split into two components, $W_g$ and $W_S$, where $W_g = W_T - W_S$, $W_g$ being the

Table I. D'Arcy flow: performance data for AMG1 ($\varepsilon \leq 0.01$).

| $Q$ | $C_g$ | $C_A$ | $W_g$ | $n$ | $\rho$ | $W_S$ | $W_T$ | $\bar{Q}$ | $C_g$ | $C_A$ | $W_g$ | $n$ | $\rho$ | $W_S$ | $W_T$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Uniform structured network A: $\varepsilon = 0.002$ | | | | | | | | Quasi-uniform unstructured network F: $\varepsilon = 0.002$ | | | | | | | |
| 16 | 1.24 | 1.50 | 19.0 | 7 | 0.0120 | 25.9 | 44.9 | 16 | 1.20 | 1.33 | 27.9 | 6 | 0.0080 | 20.9 | 48.8 |
| 32 | 1.23 | 1.59 | 22.0 | 7 | 0.0152 | 28.1 | 50.1 | 32 | 1.18 | 1.33 | 24.7 | 7 | 0.0137 | 25.6 | 50.3 |
| 64 | 1.23 | 1.65 | 24.8 | 7 | 0.0138 | 27.0 | 51.8 | 64 | 1.18 | 1.36 | 27.8 | 7 | 0.0153 | 25.2 | 53.0 |
| 96 | 1.23 | 1.67 | 27.8 | 6 | 0.0098 | 24.2 | 52.0 | 96 | 1.18 | 1.37 | 28.3 | 7 | 0.0175 | 25.6 | 53.9 |
| 128 | 1.23 | 1.68 | 29.8 | 7 | 0.0143 | 28.7 | 58.5 | 128 | 1.17 | 1.37 | 27.2 | 7 | 0.0182 | 25.7 | 52.9 |
| 160 | 1.23 | 1.68 | 31.3 | 7 | 0.0164 | 28.7 | 60.0 | 160 | 1.17 | 1.36 | 27.7 | 7 | 0.0182 | 25.4 | 53.1 |
| Uniform anisotropic network C: $\alpha = 40$; $\varepsilon = 0.06$ | | | | | | | | Unstructured non-uniform network G: $\varepsilon = 0.002$ | | | | | | | |
| 16 | 1.55 | 2.10 | 24.1 | 6 | 0.0071 | 31.9 | 56.0 | 16 | 1.19 | 1.33 | 27.5 | 7 | 0.0142 | 24.7 | 52.2 |
| 32 | 1.58 | 2.31 | 30.9 | 7 | 0.0125 | 39.3 | 70.2 | 32 | 1.21 | 1.42 | 31.6 | 8 | 0.0228 | 30.4 | 62.0 |
| 64 | 1.60 | 2.49 | 33.4 | 8 | 0.0239 | 46.7 | 80.2 | 64 | 1.21 | 1.49 | 36.9 | 7 | 0.0186 | 25.8 | 62.7 |
| 96 | 1.60 | 2.50 | 36.8 | 8 | 0.0306 | 46.8 | 83.6 | 96 | 1.21 | 1.47 | 36.5 | 8 | 0.0228 | 30.8 | 67.3 |
| 128 | 1.61 | 2.60 | 43.5 | 9 | 0.0395 | 54.1 | 97.6 | 128 | 1.20 | 1.45 | 35.7 | 8 | 0.0266 | 30.9 | 66.6 |
| 160 | 1.61 | 2.58 | 42.3 | 9 | 0.0455 | 54.1 | 96.4 | 160 | 1.19 | 1.44 | 37.1 | 8 | 0.0302 | 30.9 | 68.0 |

Table II. D'Arcy flow: performance data for AMG0 ($\beta = 0.14$).

| $Q$ | $C_g$ | $C_A$ | $W_g$ | $n$ | $\rho$ | $W_S$ | $W_T$ | $\bar{Q}$ | $C_g$ | $C_A$ | $W_g$ | $n$ | $\rho$ | $W_S$ | $W_T$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Uniform structured network A | | | | | | | | Quasi-uniform unstructured network F | | | | | | | |
| 16 | 1.27 | 1.33 | 4.9 | 12 | 0.0913 | 40.5 | 45.4 | 16 | 1.22 | 1.20 | 5.3 | 9 | 0.0428 | 31.1 | 36.4 |
| 32 | 1.29 | 1.35 | 4.2 | 13 | 0.119 | 47.9 | 52.1 | 32 | 1.21 | 1.20 | 4.8 | 12 | 0.0922 | 43.5 | 48.3 |
| 64 | 1.28 | 1.35 | 3.3 | 15 | 0.154 | 52.2 | 55.5 | 64 | 1.21 | 1.20 | 4.5 | 14 | 0.129 | 47.8 | 52.3 |
| 96 | 1.28 | 1.36 | 3.6 | 16 | 0.169 | 62.3 | 65.9 | 96 | 1.20 | 1.20 | 4.5 | 15 | 0.150 | 52.6 | 57.1 |
| 128 | 1.28 | 1.35 | 4.0 | 16 | 0.176 | 65.4 | 69.4 | 128 | 1.20 | 1.20 | 4.6 | 16 | 0.173 | 57.5 | 62.1 |
| 160 | 1.28 | 1.36 | 4.0 | 17 | 0.196 | 71.6 | 75.6 | 160 | 1.21 | 1.20 | 4.5 | 16 | 0.177 | 59.3 | 63.8 |
| Uniform anisotropic network C: $\alpha = 40$ | | | | | | | | Unstructured non-uniform network G | | | | | | | |
| 16 | 1.51 | 1.56 | 6.0 | 12 | 0.0909 | 54.2 | 60.2 | 16 | 1.24 | 1.21 | 5.5 | 10 | 0.0516 | 33.3 | 38.8 |
| 32 | 1.49 | 1.57 | 4.9 | 14 | 0.130 | 68.6 | 73.5 | 32 | 1.24 | 1.22 | 5.1 | 12 | 0.0920 | 44.2 | 49.3 |
| 64 | 1.49 | 1.60 | 4.1 | 17 | 0.186 | 76.7 | 80.8 | 64 | 1.24 | 1.22 | 4.2 | 13 | 0.115 | 44.2 | 48.4 |
| 96 | 1.50 | 1.62 | 4.5 | 17 | 0.193 | 85.5 | 90.0 | 96 | 1.23 | 1.22 | 4.4 | 15 | 0.152 | 55.1 | 59.5 |
| 128 | 1.50 | 1.62 | 4.5 | 19 | 0.227 | 101.4 | 105.9 | 128 | 1.22 | 1.22 | 4.6 | 16 | 0.174 | 58.3 | 62.9 |
| 160 | 1.50 | 1.63 | 4.6 | 20 | 0.242 | 111.2 | 115.8 | 160 | 1.22 | 1.21 | 4.8 | 18 | 0.201 | 68.4 | 73.2 |

work invested in grid generation and $W_S$ that invested in the iterative solution. Before making any direct comparisons of the performance of the two solvers, some general observations are made about the performance of each separately. It is to be remembered that each network is an inherently discrete problem in its own right; i.e. not a different approximation.

Consider first Table I, the results for the AMG1 solver. Despite the above remarks, solver performance is remarkably consistent within each network type. For all practical purposes, the convergence is $Q$-independent. Moreover, there is little difference between network types. Just one extra iteration is necessary for network G (which contains pipes varying in length by one order of magnitude) and for anisotropic network C (where orthogonal pipes differ in length by a factor 40). The strong coupling parameter $\varepsilon$ has been set at a value $\varepsilon = 0.06$; the critical value for semi-coarsening is estimated to be about 0.0125. Performance is insensitive to the precise value chosen within the range 0.0125–0.08. Note that for the anisotropic network C, the semi-coarsening results in a greater algebraic complexity. Consequently, the work burden is higher, even though the convergence rates are similar to those for other mesh types.

Consider now Table II for AMG0 ($\beta = 0.14$). In this case, $\rho$ has a more perceptible but nevertheless weak $Q$ dependence. The use of a larger grid reduction factor ($\beta = 0.5$) yield lower values (not shown) with a better, almost $Q$-independent scaling [1], although the convergence factors do not match those for AMG1. The lower $\beta$ value has been adopted here to bring the algebraic complexity more in line with that for smoothed aggregation coarsening. It also improves overall efficiency through the convergence complexity trade-off; more iterations are required but overall they prove less costly. Note that the differences between results for the anisotropic network C and the other networks are similar for this solver, i.e. slightly more sluggish. The anisotropic coarsening ('semi-coarsening') is 'hard-wired' into this coarsening algorithm through the differently defined strongly coupled neighbourhood, $N_i^l(\beta)$, as discussed in Section 3.3.2; there is no free strong coupling parameter to be set.

Finally, compare solver performances (i.e. Table I with Table II). It will be apparent that AMG1 convergence can be much more efficient than that for the AMG0, but that this advantage is usually offset by a larger grid generation cost penalty. Although AMG1 may require 50 per cent less effort in iterations, as much effort may need to be invested in grid generation. The grid generation penalty for AMG0 on the other hand is only about 10 per cent or less of the total cost, so, remarkably, the total amount of work required is similar for the two solvers, AMG1 having just a slight edge.

*5.1.2. Fick diffusion in a rectangular domain.* In this test problem Dirichlet boundary conditions are enforced on the sides of a rectangular domain (corner 1, $x = 0$, $y = 0$; corner 2, $x = 1$, $y = 0$; corner 3, $x = 1$, $y = 1$ or 20; corner 4, $x = 0$, $y = 1$ or 20). Along the boundaries of unit length the field varies linearly from zero at corners 2 and 4 to values of unity at corners 1 and 3. Thus, values of the field for the nodes lying on these boundaries are fixed while all other nodes are free. The problem is to determine the values for the field at the free nodes, which are governed by a discrete approximation to Equation (24), with the source set to zero. Four different mesh types are again examined, all assembled from triangular elements. They are (cf. Figure 8): structured homogeneous isotropic meshes B, unstructured quasi-uniform meshes F, structured homogeneous anisotropic ($\alpha = 20$) meshes D, and unstructured non-uniform meshes G, involving local refinements (of order 10:1). Unlike the previous case, the different discrete problems are different approximations to the same continuum problem, and system matrices are not necessarily $M$ matrices. In fact, mesh type G produces 'wrong sign' matrix coefficients. Again, a one-order range of mesh bandwidth is covered for each mesh type. A post-smoothing ($\nu_2 = 3$, $\nu_1 = 0$) is again employed. Results are presented in Tables III and IV for AMG1 and

Table III. Fick diffusion: performance data for AMG1 ($\varepsilon \leq 0.01$).

| $Q$ | $C_g$ | $C_A$ | $W_g$ | $n$ | $\rho$ | $W_S$ | $W_T$ | $\bar{Q}$ | $C_g$ | $C_A$ | $W_g$ | $n$ | $\rho$ | $W_S$ | $W_T$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Uniform structured mesh B: $\varepsilon = 0.002$ | | | | | | | | Quasi-uniform unstructured mesh F: $\varepsilon = 0.002$ | | | | | | | |
| 16 | 1.19 | 1.46 | 15.4 | 7 | 0.0184 | 25.5 | 40.9 | 16 | 1.16 | 1.31 | 20.7 | 7 | 0.0157 | 24.0 | 44.7 |
| 32 | 1.21 | 1.58 | 18.3 | 9 | 0.0333 | 33.2 | 51.5 | 32 | 1.16 | 1.32 | 21.2 | 8 | 0.0287 | 27.9 | 49.1 |
| 64 | 1.22 | 1.65 | 22.7 | 8 | 0.0226 | 29.7 | 52.4 | 64 | 1.17 | 1.35 | 22.8 | 9 | 0.0356 | 31.1 | 53.9 |
| 96 | 1.22 | 1.66 | 25.7 | 9 | 0.0349 | 34.6 | 60.3 | 96 | 1.17 | 1.35 | 22.5 | 9 | 0.0379 | 31.5 | 54.0 |
| 128 | 1.22 | 1.66 | 25.5 | 9 | 0.0352 | 35.4 | 60.9 | 128 | 1.17 | 1.36 | 25.4 | 9 | 0.0383 | 32.4 | 57.8 |
| 160 | 1.23 | 1.66 | 26.3 | 8 | 0.0266 | 31.3 | 57.6 | 160 | 1.17 | 1.36 | 26.0 | 9 | 0.0423 | 34.1 | 60.1 |
| Uniform anisotropic mesh D: $\alpha = 20$; $\varepsilon = 0.06$ | | | | | | | | Unstructured non-uniform mesh G: $\varepsilon = 0.002$ | | | | | | | |
| 16 | 1.52 | 2.15 | 17.7 | 7 | 0.0181 | 34.6 | 52.3 | 16 | 1.18 | 1.36 | 19.4 | 6 | 0.0059 | 19.5 | 38.9 |
| 32 | 1.61 | 2.47 | 21.3 | 8 | 0.0284 | 42.8 | 64.1 | 32 | 1.18 | 1.39 | 22.8 | 8 | 0.0242 | 28.8 | 51.6 |
| 64 | 1.65 | 2.58 | 22.8 | 9 | 0.0346 | 50.2 | 73.0 | 64 | 1.19 | 1.42 | 26.9 | 9 | 0.0361 | 32.4 | 59.3 |
| 96 | 1.66 | 2.60 | 24.6 | 9 | 0.0351 | 52.1 | 76.7 | 96 | 1.19 | 1.42 | 28.8 | 9 | 0.0384 | 32.8 | 61.6 |
| 128 | 1.68 | 2.66 | 26.7 | 9 | 0.0347 | 53.6 | 80.3 | 128 | 1.19 | 1.42 | 29.6 | 9 | 0.0417 | 33.6 | 63.2 |
| 160 | 1.68 | 2.68 | 28.0 | 9 | 0.0351 | 54.8 | 82.9 | 160 | 1.19 | 1.43 | 29.6 | 9 | 0.0427 | 33.6 | 63.2 |

Table IV. Fick diffusion: performance data for AMG0 ($\beta = 0.14$).

| $Q$ | $C_g$ | $C_A$ | $W_g$ | $n$ | $\rho$ | $W_S$ | $W_T$ | $\bar{Q}$ | $C_g$ | $C_A$ | $W_g$ | $n$ | $\rho$ | $W_S$ | $W_T$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Uniform structured mesh B | | | | | | | | Quasi-uniform unstructured mesh F | | | | | | | |
| 16 | 1.19 | 1.27 | 4.5 | 9 | 0.0427 | 29.9 | 34.4 | 16 | 1.18 | 1.18 | 4.0 | 8 | 0.0303 | 25.3 | 29.3 |
| 32 | 1.23 | 1.30 | 3.4 | 14 | 0.131 | 50.4 | 53.8 | 32 | 1.19 | 1.19 | 4.2 | 12 | 0.0982 | 40.5 | 44.7 |
| 64 | 1.24 | 1.31 | 3.2 | 17 | 0.184 | 60.0 | 63.2 | 64 | 1.20 | 1.20 | 4.0 | 16 | 0.163 | 53.8 | 57.8 |
| 96 | 1.24 | 1.31 | 3.3 | 18 | 0.204 | 66.2 | 69.5 | 96 | 1.20 | 1.20 | 4.1 | 17 | 0.180 | 59.9 | 64.0 |
| 128 | 1.24 | 1.32 | 3.5 | 19 | 0.224 | 72.4 | 75.9 | 128 | 1.20 | 1.21 | 4.4 | 17 | 0.197 | 60.5 | 64.9 |
| 160 | 1.24 | 1.32 | 3.3 | 19 | 0.231 | 75.9 | 79.2 | 160 | 1.21 | 1.21 | 4.0 | 18 | 0.206 | 66.8 | 70.8 |
| Uniform anisotropic mesh D: $\alpha = 20$ | | | | | | | | Unstructured non-uniform mesh G | | | | | | | |
| 16 | 1.35 | 1.46 | 3.5 | 11 | 0.0716 | 38.2 | 41.7 | 16 | 1.17 | 1.19 | 3.7 | 6 | 0.0099 | 19.4 | 23.1 |
| 32 | 1.41 | 1.49 | 3.6 | 14 | 0.125 | 57.6 | 61.2 | 32 | 1.18 | 1.19 | 3.8 | 11 | 0.0629 | 34.4 | 38.2 |
| 64 | 1.45 | 1.55 | 3.6 | 17 | 0.182 | 76.7 | 80.3 | 64 | 1.19 | 1.20 | 4.0 | 14 | 0.131 | 47.1 | 51.1 |
| 96 | 1.47 | 1.56 | 3.7 | 18 | 0.204 | 83.4 | 87.1 | 96 | 1.20 | 1.20 | 3.8 | 15 | 0.157 | 51.1 | 54.9 |
| 128 | 1.46 | 1.57 | 3.8 | 19 | 0.231 | 92.6 | 96.4 | 128 | 1.20 | 1.20 | 4.0 | 16 | 0.174 | 56.5 | 60.5 |
| 160 | 1.48 | 1.58 | 4.0 | 20 | 0.245 | 102.2 | 106.2 | 160 | 1.20 | 1.21 | 4.2 | 17 | 0.185 | 62.9 | 67.1 |

AMG0 solvers respectively, which may be compared directly with previous results of Tables I and II.

The performance tables are very similar to the previous results for networks. Indeed, remarks made for the network results apply equally well to these results for diffusion. Just one or two extra (or fewer) cycles may be required to achieve the convergence. In the case of the unstructured, non-uniform mesh G, the system matrix had some wrong sign off-diagonal entries but despite this the convergence is comparable with that for the other mesh types. For anisotropic mesh D, the convergences are again broadly similar to the previous results for networks, and the work burdens are similarly higher due to the increased complexity associated with preferred, directional coarsening.

### 5.2. *Vector field systems*

*5.2.1. Displacement fields for plane strain elasticity*. This application is the calculation of a strain field in a square/rectangular section of an elastic material with a Poisson's ratio of 0.3. The four corners of the domain are subjected to fixed displacements directed outward along the diagonals from the centre of area. The problem is to determine the vector displacement field, $u_i$, and hence the resulting change in shape of the section. The two components of the displacement are determined from the discrete approximations to Equations (28) and (29). These are again based on the four finite element mesh types described above (i.e. types B, F, D and G) and a similar range of mesh bandwidths is examined. The system matrices differ from the previous cases, being block structured, with large entries in the off-diagonal blocks representing the interaction between the field components, $u_i$, as described in Section 4.1.2. While for single-field applications the problems of inhomogeneity and anisotropy are addressed automatically by the algebraic coarsening scheme (i.e. implicit preferential intra-field coarsening described above), for multi-field applications any adverse influence of the anisotropy on the inter-field coupling may not be addressed. This could result in poor smoothing and a degraded overall performance. The number of post smoothing sweeps is thus increased ($v_2 = 5$, $v_1 = 0$).

The performance data for this case are presented in Tables V and VI for AMG1 and AMG0 solvers respectively. They may be compared directly with corresponding sets in Tables I, II, III and IV. The most noticeable difference is the slower convergence, which is to be expected. Also to be noted is the larger grid reduction factor ($\beta = 0.5$) that has been used for the AMG0 solver. Attempts to use the lower values ($\beta = 0.14$) resulted in stagnated convergence for the higher resolution meshes ($Q > 96$). However, convergence proved to be robust for the AMG0 ($\beta = 0.5$) solver and for the AMG1 solver. Like previous cases, convergence factors are almost mesh-independent. Since convergence is more sluggish for this tougher problem, more effort is required in the solution phase, and since the AMG1 solver is more efficient in this phase, it begins to show a better overall efficiency. Thus, for all but the anisotropic mesh and for the low-resolution meshes, fewer work units are required for the AMG1 solver to achieve convergence.

With regard to the comparative performance for the different mesh types, this followed the same general pattern as for scalar field problems, i.e. virtually mesh-independent. If

Table V. Plane strain elasticity performance data for AMG1.

| $Q$ | $C_g$ | $C_A$ | $W_g$ | $n$ | $\rho$ | $W_S$ | $W_T$ | $\bar{Q}$ | $C_g$ | $C_A$ | $W_g$ | $n$ | $\rho$ | $W_S$ | $W_T$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Uniform structured mesh B: $\varepsilon = 0.002$ | | | | | | | | Quasi-uniform unstructured mesh F: $\varepsilon = 0.002$ | | | | | | | |
| 16 | 1.25 | 1.45 | 21.0 | 10 | 0.0514 | 34.6 | 55.5 | 16 | 1.20 | 1.33 | 25.1 | 10 | 0.0540 | 33.7 | 58.8 |
| 32 | 1.25 | 1.54 | 27.5 | 11 | 0.0733 | 37.3 | 64.8 | 32 | 1.20 | 1.36 | 24.9 | 11 | 0.0740 | 35.0 | 59.9 |
| 64 | 1.24 | 1.56 | 32.0 | 12 | 0.0837 | 42.4 | 74.4 | 64 | 1.19 | 1.39 | 29.2 | 12 | 0.0912 | 39.3 | 68.5 |
| 96 | 1.25 | 1.62 | 42.8 | 12 | 0.0985 | 44.6 | 87.4 | 96 | 1.18 | 1.39 | 30.1 | 12 | 0.0933 | 39.8 | 69.9 |
| 128 | 1.23 | 1.58 | 36.3 | 12 | 0.0899 | 47.6 | 83.9 | 128 | 1.18 | 1.39 | 33.7 | 12 | 0.0943 | 52.3 | 86.0 |
| 160 | 1.23 | 1.59 | 37.0 | 12 | 0.0915 | 59.6 | 96.6 | 160 | 1.17 | 1.39 | 36.7 | 12 | 0.0965 | 63.2 | 99.9 |
| Uniform anisotropic mesh D: $\alpha = 20$ $\varepsilon = 0.06$ | | | | | | | | Unstructured non-uniform mesh G: $\varepsilon = 0.002$ | | | | | | | |
| 16 | 1.70 | 2.24 | 24.4 | 8 | 0.0304 | 43.2 | 67.6 | 16 | 1.23 | 1.38 | 25.8 | 10 | 0.0515 | 33.5 | 59.3 |
| 32 | 1.72 | 2.44 | 25.3 | 10 | 0.0520 | 52.6 | 77.9 | 32 | 1.23 | 1.43 | 27.9 | 12 | 0.0827 | 39.4 | 67.3 |
| 64 | 1.70 | 2.45 | 28.6 | 10 | 0.0556 | 56.4 | 85.0 | 64 | 1.22 | 1.45 | 32.7 | 12 | 0.0910 | 40.1 | 72.8 |
| 96 | 1.70 | 2.50 | 32.3 | 12 | 0.0862 | 70.8 | 103.1 | 96 | 1.22 | 1.46 | 36.6 | 12 | 0.0967 | 42.6 | 79.2 |
| 128 | 1.70 | 2.51 | 32.1 | 11 | 0.0795 | 83.5 | 115.6 | 128 | 1.21 | 1.46 | 36.8 | 12 | 0.0980 | 44.7 | 81.5 |
| 160 | 1.70 | 2.50 | 35.4 | 12 | 0.0873 | 95.3 | 130.6 | 160 | 1.21 | 1.47 | 38.1 | 12 | 0.0994 | 61.5 | 99.6 |

Table VI. Plane strain elasticity: performance data for AMG0 ($\beta = 0.5$).

| $Q$ | $C_g$ | $C_A$ | $W_g$ | $n$ | $\rho$ | $W_S$ | $W_T$ | $\bar{Q}$ | $C_g$ | $C_A$ | $W_g$ | $n$ | $\rho$ | $W_S$ | $W_T$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Uniform structured mesh B | | | | | | | | Quasi-uniform unstructured mesh F | | | | | | | |
| 16 | 1.92 | 1.78 | 3.69 | 12 | 0.0953 | 68.2 | 71.9 | 16 | 1.92 | 1.87 | 4.5 | 10 | 0.0487 | 51.4 | 55.9 |
| 32 | 1.98 | 1.82 | 3.60 | 17 | 0.185 | 94.8 | 98.4 | 32 | 1.98 | 1.93 | 4.4 | 14 | 0.133 | 78.7 | 83.1 |
| 64 | 1.99 | 1.81 | 3.59 | 22 | 0.277 | 145.8 | 149.4 | 64 | 2.00 | 1.95 | 4.5 | 19 | 0.217 | 125.9 | 130.4 |
| 96 | 2.00 | 1.88 | 3.86 | 24 | 0.306 | 182.8 | 186.7 | 96 | 2.00 | 1.96 | 4.6 | 20 | 0.246 | 156.7 | 161.3 |
| 128 | 2.00 | 1.81 | 3.65 | 26 | 0.339 | 232.1 | 235.7 | 128 | 2.00 | 1.96 | 5.3 | 22 | 0.274 | 219.4 | 224.7 |
| 160 | 2.00 | 1.88 | 4.49 | 26 | 0.345 | 267.2 | 271.7 | 160 | 2.00 | 1.97 | 5.8 | 23 | 0.286 | 265.8 | 271.6 |
| Uniform anisotropic mesh D; $\alpha = 20$ | | | | | | | | Unstructured non-uniform mesh G | | | | | | | |
| 16 | 1.92 | 1.82 | 3.8 | 13 | 0.109 | 72.3 | 76.1 | 16 | 1.92 | 1.83 | 4.9 | 11 | 0.0639 | 64.2 | 69.1 |
| 32 | 1.98 | 1.90 | 3.3 | 16 | 0.167 | 87.6 | 90.9 | 32 | 1.98 | 1.92 | 4.2 | 14 | 0.123 | 76.4 | 80.6 |
| 64 | 1.99 | 1.95 | 3.8 | 18 | 0.212 | 120.6 | 124.4 | 64 | 1.99 | 1.95 | 4.4 | 17 | 0.186 | 115.4 | 119.8 |
| 96 | 2.00 | 1.97 | 4.1 | 19 | 0.231 | 148.0 | 152.1 | 96 | 2.00 | 1.96 | 4.7 | 19 | 0.219 | 148.6 | 153.3 |
| 128 | 2.00 | 1.97 | 4.0 | 21 | 0.264 | 212.4 | 216.4 | 128 | 2.00 | 1.96 | 4.6 | 20 | 0.238 | 179.5 | 184.1 |
| 160 | 2.00 | 1.97 | 3.8 | 22 | 0.272 | 234.5 | 238.3 | 160 | 2.00 | 1.97 | 5.4 | 20 | 0.249 | 207.4 | 212.8 |

anything, performance for the AMG0 solver is slightly more sluggish for the structured, uniform mesh types.

To summarize, both solvers perform effectively albeit slightly less efficiently for this vector field despite the strong coupling between vector components.

### *5.3. Coupled scalar–vector field systems*

As an example of a coupled scalar–vector field system, the hydrodynamics of a viscous fluid has been chosen (Equations (34) and (35)). With regard to the general problem of solver bandwidth, the non-linear advective forces in Equation (34) should not be an issue. Being first-order they should be interpolated/restricted consistently in both AMG0 and AMG1. It is the dominance of the second-order terms for low-Reynolds number Stokes problems that present the more testing challenge to AMG0 in view of its zero-order transfer operators. For this reason, problems with small or negligible advective forces have been selected. This choice also conveniently avoids the presentational problem of dealing with non-linear iteration sequences, only one call on the linear solver being necessary. Two problems have been chosen: a shear-driven viscous flow in a square cavity and a uni-directional flow in a rectangular duct.

*5.3.1. Shear-driven flow in a square cavity.* Three sides of the cavity are none-slip walls, while the fourth has a prescribed constant velocity. Just one boundary node has a fixed constant pressure; all other pressure nodes are free. A viscous flow ($Re = 3$) is chosen to ensure the above requirement for relatively weak advective forces is satisfied. The discrete approximations are derived straightforwardly using the finite volume method described in Section 4.3; however, the involvement of both first and second derivatives requires the introduction of a special interpolation based on the physics of the problem (see for example Reference [7]). Thus, the velocity in the continuity equation is an element velocity that is expressed as a linear function of the nodal velocities and pressures. As in the previous application, the coupled discrete system is block structured, with diagonal blocks $\mathbf{A}_{qq}$ representing the advection–diffusion operators in the case of the velocity equations ($\mathbf{x}_q = \mathbf{u}_i$) and a Poisson-type operator in the case of the pressure equation ($\mathbf{x}_q = \mathbf{p}$), while the off-diagonal blocks $\mathbf{A}_{pq}$ represent discrete gradient operators in the case of the velocity equations and a discrete divergence operator in the case of the pressure equations.

The mesh types adopted are types B, F, G and E. Mesh parameters for type G are identical to those used in the previous applications, i.e. giving a one-order local refinement at each of the four corners. For the structured orthogonal mesh type E, there is a refinement by a factor 20 to both the top and the right sides, which necessarily involves elements of aspect ratio up to $\alpha = 20$. Again, as in the vector field application, in recognition of the possible deleterious influence of the off-diagonal blocks on smoothing, the number of post-smoothing sweeps is set moderately high, $v_2 = 5$ ($v_1 = 0$).

The performance data is presented in Tables VII and VIII, and these may again be compared directly with the previous tables (providing caution is exercised with regard to comparisons of the work done). For a given mesh bandwidth, coupled scalar–vector fields require more storage and hence are more likely to have a higher overhead in memory management when core storage is limited and virtual memory transfers are required. For example, note the misleading decrease in work units for an increase in bandwidth from $Q = 128$ to $Q = 160$. This is simply a reflection of a more costly work unit due to such machine overheads. In making comparisons of work done, therefore, it is important to only compare like with like (with respect to both problem application and mesh bandwidth).

Convergence data on the other hand are directly comparable regardless of the machine efficiency. Despite differences in the system matrix and, in particular, in the nature of the

Table VII. Navier–Stokes: driven cavity: performance data for AMG1.

| $Q$ | $C_g$ | $C_A$ | $W_g$ | $n$ | $\rho$ | $W_S$ | $W_T$ | $\bar{Q}$ | $C_g$ | $C_A$ | $W_g$ | $n$ | $\rho$ | $W_S$ | $W_T$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Uniform structured mesh B: $\varepsilon = 0.002$ | | | | | | | | Quasi-uniform unstructured mesh F: $\varepsilon = 0.002$ | | | | | | | |
| 16 | 1.26 | 1.45 | 18.0 | 10 | 0.0607 | 34.8 | 52.8 | 16 | 1.21 | 1.32 | 20.7 | 10 | 0.0464 | 33.0 | 53.7 |
| 32 | 1.28 | 1.53 | 21.6 | 11 | 0.0664 | 39.4 | 61.0 | 32 | 1.22 | 1.36 | 21.4 | 10 | 0.0590 | 32.5 | 53.9 |
| 64 | 1.30 | 1.57 | 27.1 | 11 | 0.0733 | 42.5 | 69.6 | 64 | 1.22 | 1.36 | 23.7 | 12 | 0.0816 | 40.7 | 64.4 |
| 96 | 1.30 | 1.60 | 30.1 | 11 | 0.0747 | 49.0 | 79.1 | 96 | 1.22 | 1.37 | 25.5 | 12 | 0.0894 | 42.0 | 67.5 |
| 128 | 1.32 | 1.68 | 42.3 | 11 | 0.0772 | 72.3 | 114.6 | 128 | 1.22 | 1.37 | 27.6 | 13 | 0.113 | 65.8 | 93.4 |
| 160 | 1.31 | 1.61 | 8.62 | 12 | 0.0949 | 40.0 | 48.6 | 160 | 1.21 | 1.36 | 7.80 | 13 | 0.108 | 40.6 | 48.4 |
| $\bar{Q}$ | | | | | | | | | | | | | | | |
| Uniform anisotropic mesh E: $\alpha = 20$ $\varepsilon = 0.06$ | | | | | | | | Unstructured non-uniform mesh G: $\varepsilon = 0.002$ | | | | | | | |
| 16 | 1.40 | 1.70 | 16.8 | 7 | 0.0117 | 24.7 | 41.4 | 16 | 1.22 | 1.35 | 19.4 | 8 | 0.0275 | 25.3 | 44.7 |
| 32 | 1.41 | 1.74 | 17.9 | 8 | 0.0207 | 27.9 | 45.8 | 32 | 1.24 | 1.40 | 23.3 | 10 | 0.0524 | 32.9 | 56.2 |
| 64 | 1.46 | 1.91 | 26.6 | 9 | 0.0439 | 35.3 | 61.9 | 64 | 1.24 | 1.40 | 24.1 | 11 | 0.0728 | 37.5 | 61.6 |
| 96 | 1.47 | 1.96 | 30.4 | 10 | 0.0570 | 47.9 | 78.2 | 96 | 1.24 | 1.41 | 28.0 | 12 | 0.0851 | 43.0 | 71.7 |
| 128 | 1.47 | 1.96 | 31.2 | 12 | 0.0973 | 65.1 | 96.4 | 128 | 1.24 | 1.42 | 31.5 | 12 | 0.0958 | 64.3 | 95.8 |
| 160 | 1.47 | 1.95 | 11.2 | 12 | 0.0899 | 38.9 | 50.0 | 160 | 1.24 | 1.42 | 8.00 | 13 | 0.103 | 38.3 | 46.3 |

inter-field coupling, performance is similar to the previous vector field applications. The convergence factors are of a similar magnitude, are largely mesh-independent or, at worst, only weakly mesh-dependent. Note also that the AMG0 data is for a target $\beta$ value of $\beta = 0.14$; unlike the previous case, reliable convergence was obtained even for low $\beta$ values.

*5.3.2. Flow in a rectangular duct.* The above tests for discretizations based on type D meshes demonstrate that rates of convergence for anisotropic problems can be just as rapid as those for isotropic problems. Note that the degree of anisotropy was of order $10^2$–$10^3$ ($\sim \alpha^2$ for these largely elliptic problems). In this final test set, solver performance is examined for coupled pressure–velocity systems of more extreme anisotropy.

High aspect ratio type D meshes are again chosen as the means of generating large anisotropies. The problem addressed is that of a pressure-driven, lamina flow along a duct of rectangular cross-section ($Re = 0.174$). The length to half-width ratio is 128, i.e. the duct is 32 hydraulic diameters in length. Both inlet and outlet (left and right respectively) are prescribed pressure boundaries. A solid no-slip wall and a free slip symmetry axis constitute the top and bottom boundaries. Three discretizations are examined that have aspect ratios of 16, 64 and 256 (anisotropies of up to 256, 4096 and 65536 respectively), each having 24576 degrees of freedom (axial and transverse bandwidths as listed in Table IX). In all of these highly anisotropic tests, smoothing has been improved using an ILU0 preconditioner, similar to that described by Raw *et al.* [9].

The performance data of Table IX confirm that efficient convergence can be obtained for coupled field problems of extreme anisotropy. Convergence rates are broadly in line with those for previous test problems. Note however the cost of smoothing. The amount of processing

Table VIII. Navier–Stokes: driven cavity: performance data AMG0 ($\beta = 0.14$).

| $Q$ | $C_g$ | $C_A$ | $W_g$ | $n$ | $\rho$ | $W_S$ | $W_T$ | $\bar{Q}$ | $C_g$ | $C_A$ | $W_g$ | $n$ | $\rho$ | $W_S$ | $W_T$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Uniform structured mesh B | | | | | | | | Quasi-uniform unstructured mesh F | | | | | | | |
| 16 | 1.23 | 1.25 | 5.5 | 13 | 0.114 | 41.9 | 47.4 | 16 | 1.19 | 1.18 | 6.4 | 11 | 0.0768 | 32.4 | 38.8 |
| 32 | 1.25 | 1.26 | 4.7 | 20 | 0.240 | 62.8 | 67.5 | 32 | 1.20 | 1.18 | 5.9 | 18 | 0.203 | 52.3 | 58.2 |
| 64 | 1.26 | 1.27 | 5.1 | 25 | 0.328 | 84.9 | 90.0 | 64 | 1.20 | 1.18 | 6.1 | 23 | 0.294 | 70.2 | 76.3 |
| 96 | 1.26 | 1.27 | 5.1 | 28 | 0.369 | 100.0 | 105.4 | 96 | 1.21 | 1.19 | 6.3 | 26 | 0.341 | 84.2 | 90.5 |
| 128 | 1.26 | 1.27 | 5.4 | 31 | 0.401 | 144.6 | 150.0 | 128 | 1.21 | 1.19 | 6.2 | 28 | 0.370 | 107.6 | 113.8 |
| 160 | 1.26 | 1.26 | 5.4 | 33 | 0.422 | 99.5 | 101.4 | 160 | 1.21 | 1.18 | 2.4 | 29 | 0.383 | 83.8 | 86.2 |
| $\bar{Q}$ | | | | | | | | | | | | | | | |
| Uniform anisotropic mesh E | | | | | | | | Unstructured non-uniform mesh G | | | | | | | |
| 16 | 1.32 | 1.35 | 2.9 | 6 | 0.0066 | 18.4 | 21.3 | 16 | 1.19 | 1.18 | 5.8 | 10 | 0.0607 | 28.7 | 34.5 |
| 32 | 1.34 | 1.36 | 2.9 | 10 | 0.057 | 30.7 | 33.5 | 32 | 1.21 | 1.19 | 5.5 | 14 | 0.126 | 40.7 | 46.2 |
| 64 | 1.36 | 1.37 | 3.1 | 16 | 0.169 | 54.5 | 57.6 | 64 | 1.20 | 1.18 | 6.0 | 19 | 0.229 | 57.3 | 63.3 |
| 96 | 1.37 | 1.38 | 3.3 | 19 | 0.219 | 71.4 | 74.7 | 96 | 1.20 | 1.19 | 6.4 | 22 | 0.271 | 70.2 | 76.6 |
| 128 | 1.37 | 1.38 | 3.3 | 21 | 0.256 | 95.1 | 98.4 | 128 | 1.20 | 1.18 | 6.2 | 23 | 0.299 | 96.5 | 102.7 |
| 160 | 1.37 | 1.38 | 1.22 | 24 | 0.309 | 73.8 | 75.1 | 160 | 1.20 | 1.19 | 1.9 | 25 | 0.325 | 72.8 | 74.7 |

involved in each application of the smoother (the absolute cost of each work unit) is large. As many as $\nu_2 = 40$ ILU0-preconditioned GCR iterations are required for each smoothing operation. Without this investment the AMG0 solver fails completely at large anisotropies ($\rho > 0.95$ for $\nu_2 < 40$, $\alpha > 16$). It can succeed for low bandwidth meshes ($Q < 64$) or if some underrelaxation of the pressure–velocity coupling is exploited in the smoothing, but this is not guaranteed. It has to be assumed that without a better smoother, the AMG0 solver is not suitable for coupled-field problems of extreme anisotropy.

Table IX. Navier–Stokes: pressure-driven pipe flow: $Re = 0.174$: performance data for structured type D meshes of high anisotropy.

| $\alpha$ | $Q_x$ | $Q_y$ | $C_g$ | $C_A$ | $W_g$ | $n$ | $\rho$ | $W_S$ | $W_T$ |
|---|---|---|---|---|---|---|---|---|---|
| AMG1, $\varepsilon = 0.08$; $\nu_1 = 0$; $\nu_2 = 40$ | | | | | | | | | |
| 16 | 256 | 32 | 1.74 | 2.14 | 1.13 | 8 | 0.0219 | 58.3 | 59.4 |
| 64 | 128 | 64 | 1.85 | 2.37 | 1.48 | 12 | 0.0862 | 94.4 | 95.9 |
| 256 | 64 | 128 | 1.92 | 2.45 | 2.17 | 11 | 0.0727 | 91.3 | 93.5 |
| AMG0, $\beta = 0.5$; $\nu_1 = 0$; $\nu_2 = 40$ | | | | | | | | | |
| 16 | 256 | 32 | 1.98 | 1.94 | 0.27 | 17 | 0.193 | 145.1 | 145.3 |
| 64 | 128 | 64 | 1.98 | 1.97 | 0.25 | 20 | 0.247 | 166.3 | 166.5 |
| 256 | 64 | 128 | 1.98 | 1.97 | 0.25 | 23 | 0.294 | 187.3 | 187.5 |

Table X. Navier–Stokes pressure-driven pipe flow, $Re = 0.174$: performance data for structured type D meshes of high anisotropy

| $\alpha$ | $Q_x$ | $Q_y$ | $C_g$ | $C_A$ | $W_g$ | $n$ | $\rho$ | $W_S$ | $W_T$ |
|---|---|---|---|---|---|---|---|---|---|
| AMG1, $\varepsilon = 0.08$; $\nu_1 = 0$; $\nu_2 = 5$ | | | | | | | | | |
| 16 | 256 | 32 | 1.76 | 2.17 | 16.24 | 31 | 0.401 | 364.3 | 280.6 |
| 64 | 128 | 64 | 1.84 | 2.34 | 19.71 | 35 | 0.452 | 322.5 | 342.2 |
| 256 | 64 | 128 | 1.94 | 2.44 | 31.24 | 38 | 0.488 | 370.3 | 401.5 |

The AMG1 solver is more robust. A reliable, full convergence is achieved with fewer smoother sweeps ($\rho < 0.5$ for $\nu_2 > 16$, $\alpha \leq 256$) without any requirement for an underrelaxation of the inter-field coupling. With underrelaxation fewer smoother sweeps are possible. Any amount of underrelaxation may be used up to 100 per cent, (with an 80 per cent underrelaxation, $\rho < 0.5$ for $\nu_2 > 4$, $\alpha \leq 256$). Convergence rates, however, are lower than those for more isotropic problems (Table X).

Clearly for coupled pressure–velocity fields, problems of extreme anisotropy are difficult when smoothers are based on simple local relaxation. As in geometric multigrid, more complex smoothing operators may be required.

## 6. CONCLUSIONS

For single-field systems, both AMG1 and AMG0 solvers provide the expected multigrid performance. Convergence is, for all practical purposes mesh independent, i.e. independent of mesh bandwidth and of mesh inhomogeneity/anisotropy, even for highly anisotropic meshes assembled from elements of high aspect ratio.

For coupled field systems both solvers can also provide multigrid performance as long as the smoothers are effective. This may not be the case for coupled pressure–velocity systems when the problems are highly anisotropic (e.g. uniform assemblies of elements with extreme aspect ratios, $\alpha \gg 10$). For such difficult cases, the AMG1 solver proves to be more robust. Where the AMG0 solver fails, more expensive smoothers may be required, similar to those used in geometric multigrid algorithms.

For both single field and coupled field problems, AMG0 is less efficient than AMG1 in solution; convergence rate can be a factor two slower. The scaling of the AMG0 convergence rates can also be more discernibly mesh-dependent. However, for the range of mesh bandwidth examined, differences in total computing time are less significant because AMG0 is much more efficient in grid generation. The grid generation cost of AMG0 is only a small fraction of the total work burden, whereas for AMG1 it can be as high as that for the solution phase for full convergence (to machine accuracy).

Since solution phase costs will usually dominate the overall cost of AMG0, it benefits most from any relaxation of the demanded level of convergence. Most practical levels ($\sim Q^{-p}$; $p$ being the order of the truncation error) are not as tight as those demanded here. This would make AMG0 the preferred choice as a linear approximation solver in non-linear solution algorithms for Navier–Stokes problems, where just modest reductions in residual errors are demanded for each update of the approximation. Unfortunately, it is just such problems that are sometimes discretized using large assemblies of elements of extreme aspect ratio.

## REFERENCES

1. Webster R. Efficient algebraic multigrid solvers with elementary restriction and prolongation. *International Journal for Numerical Methods in Fluids* 1998; **28**: 317–336.
2. Mandel J, Brezina M, Vanáek P. Energy optimisation of algebraic multigrid bases. University of Colorado at Denver, Center for Computational Mathematics, UCD/CCM Report 125, February. Revised October, 1998; 1998.
3. Vanáek P, Mandel J, Brezina M. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. In *NASA Conference Publication 3339 Part 2*, *Copper Mountain Conference on Multigrid Methods*, Melson ND, Manteuffel TA, Cormack SF (eds). National Aeronautics and Space Administration: Langley Research Center, Hampton, Virginia, 1995; 721–735.
4. Hemker PW. On the order of prolongations and restrictions in multigrid procedures. *Journal of Computer Applications and Mathematics* 1990; **32**: 423–429.
5. Brandt A. Multi-level adaptive techniques (MLAT) for partial differential equations: ideas and software. In *Proceedings of the Symposium on Mathematical Software*, Rice J (ed.). Academic: New York, 1977; 277–318.
6. Ruge J, Stüben K. Algebraic multigrid. In *Multigrid Methods*, *Frontiers in Applied Mathematics*, vol. 5, Cormick S (ed.). SIAM: Philadelphia, PA, 1987; 73–130.
7. Koobus B, Lallemand M-H, Dervieux A. Unstructured volume-agglomeration MG: solution of Poisson equation. *International Journal for Numerical Methods in Fluids* 1994; **18**: 27–42.
8. Webster R. An algebraic multigrid solver for Navier–Stokes problems. *International Journal for Numerical Methods in Fluids* 1994; **18**: 761–780.
9. Raw M. Robustness of coupled algebraic multigrid for the Navier–Stokes equations. In *Proceedings of the 34th Aerospace Sciences Meeting and Exhibit*, Reno, NV, 1996. AIAA 96-0297.