A NOTE ON MULTI-BLOCK RELAXATION SCHEMES FOR MULTIGRID SOLVERS

JIM E. JONES*
ICASE
NASA Langley Research Center
Hampton, VA 23681-0001
jjones@icase.edu

N. DUANE MELSON
NASA Langley Research Center
Hamtpon, VA 23681-0001
n.d.melson@larc.nasa.gov

Abstract

Efficient and robust multigrid solvers for anisotropic problems typically use either semi-coarsened grids or implicit smoothers - line relaxation in 2D and plane relaxation in 3D. However, both of these may be difficult to implement in codes using multi-block structured grids where there may be no natural definition of a global 'line' or 'plane'. These multi-block structured grids are often used in fluid dynamic applications to capture complex geometries and/or to facilitate parallel processing. In this paper, we investigate the performance of multigrid algorithms using implicit smoothers *within the blocks* of a such a grid. By looking at a model problem, the 2-D anisotropic diffusion equation, we show that true multigrid efficiency is achieved only when the block sizes are proportional to the strength of the anisotropy. Further, the blocks must overlap and the size of the overlap must again be proportional to the strength of the anisotropy.

* This research was supported in part by the National Aeronautics and Space Administration under NASA Contract No. NAS1-19480 while the first author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681-0001

1. Introduction and Motivation.

To illustrate the main issues involved in designing efficient multigrid solvers for anisotropic problems, consider the problem of solving the partial differential equation

$$\epsilon \frac{\partial^2 u(x,y)}{\partial x^2} + \frac{\partial^2 u(x,y)}{\partial y^2} = f(x,y), \tag{1}$$

along with some suitable boundary condition on a rectangular domain $\Omega \subset \mathcal{R}^2$ whose boundaries are parallel to the xy coordinates. This PDE can be discretized by placing a uniform grid (with mesh size h) over the domain and using second-order finite difference approximations for the derivatives. We associate to each vertex in the grid an index (I, J) in the usual manner and let $U^h_{I,J}$ and $F^h_{I,J}$ denote the discrete approximations to the continuous variables u and f appearing in equation(1). The discrete equations then have the form

$$\frac{1}{h^2}\left[\epsilon U^h_{I+1,J} + \epsilon U^h_{I-1,J} + U^h_{I,J+1} + U^h_{I,J-1} - 2(1+\epsilon)U^h_{I,J}\right] = F^h_{I,J}. \tag{2}$$

Note that here we have considered an anisotropic PDE discretized on a grid with equal mesh size in the x and y directions. The same discrete equations arise when an isotropic PDE ($\epsilon = 1$) is discretized on grid with mesh size h in the y-direction and $h/\sqrt{\epsilon}$ in the x-direction.

Multigrid methods rely on two processes: a relaxation process to reduce high-frequency error and a coarse grid correction process to reduce low-frequency error. When standard coarsening is used (i.e. coarse grids are obtained from fine grids by deleting every other line in each coordinate direction) error components that are oscillatory in either coordinate direction must be effectively reduced by relaxation as these components cannot be represented on the coarser grid. Using the tool of local mode analysis on the above discrete equations with $0 < \epsilon \ll 1$, one can show (see [1]) that point Gauss-Seidel relaxation (or more generally, any point-wise relaxation process) does not efficiently reduce error components that are oscillatory in the x-direction but smooth in the y-direction. One solution is to accept this limitation of the relaxation process and coarsen the grid only in the y-direction. The other solution is line relaxation in the y-direction. Here all unknowns sharing the same I index are updated simultaneously, and local mode analysis shows that this relaxation process effectively reduces all high-frequency error components. Either solution, semi-coarsening or line relaxation, yield multigrid algorithms which are efficient for small ϵ.

In the field of Computational Fluid Dynamics (CFD), discrete formulations of the equations representing the flow around aerodynamic bodies are solved. These equations are solved at discrete points defined by a previously generated body-fitted grid. The body fitted grid can be what is referred to as a structured or an unstructured grid. For unstructured grids, the relative position of the grid points is explicitly stored and must be referred to by the flow solver when calculating the grid metric quantities. For structured grids, the relative placement of the grid points is implicitly known by the relative position of the data referring to the points in the program's data arrays. Unstructured grids offer greater flexibility to fit complex geometries but carry the overhead of needing

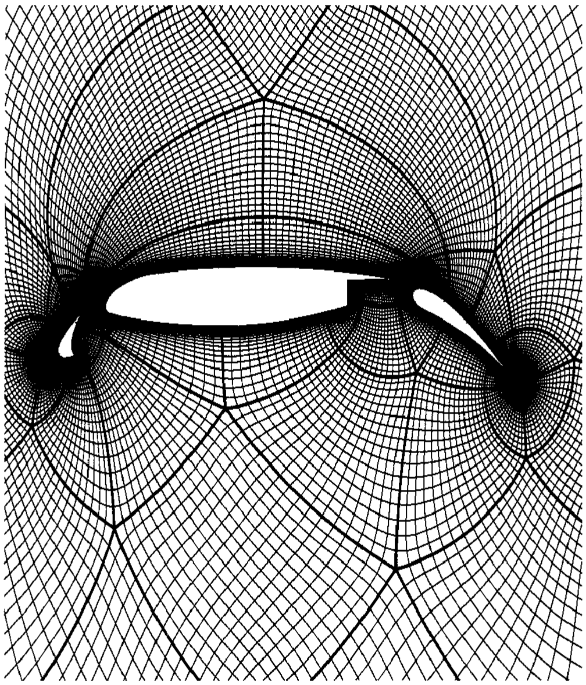

FIG. 1. *Multi-bock grid for a multi-element airfoil.*

to carry explicit information about the relative position of each point to its neighbors. Structured grids are inherently more efficient computationally but are limited in their ability to fit complex geometries. Block-structured (or multi-block) grids can be used to provide greater geometric flexibility for structured grids.

In a block-structured grid, both semi-coarsening and line relaxation may be difficult to implement as there may be no natural definition of a global line. An example of such a grid is shown in figure(1). Further, if the blocks are stored on different processors in a parallel computer, line-relaxation will require the solution of a tridiagonal system which is distributed across, perhaps, many processors. These potential difficulties led us to investigate the use of line-relaxation only *within the blocks* of the grid. We will see in both the numerical experiments and the analysis, that line-relaxation within the blocks is generally not enough to obtain efficient multigrid solvers. To obtain the same convergence factors as obtained using point-wise Gauss-Seidel relaxation of the isotropic problem, the blocks must overlap with their neighboring blocks and both the size of the blocks and the overlap must be proportional to the strength of the anisotropy.

2. Numerical Experiments. To investigate the effect of anisotropy strength, block size and overlap on multigrid efficiency, we construct the following 2D test case. Using a uniform grid of size $(2^k+1)\times(2^k+1), k$ an integer, and mesh size $h=1$; the PDE in equation(1) is discretized using finite differences as in equation(2). Dirichlet boundary conditions are assumed so that the value of the solution u at extreme grid points, those with i or j index equal to 1 or 2^k+1, is given. This leaves $(2^k-1)\times(2^k-1)$ interior vertices where an approximate solution $U^h_{i,j}$ must be calculated. We divide these vertices into regular blocks of, as much as possible, equal size: $M \times M$. Note that if M

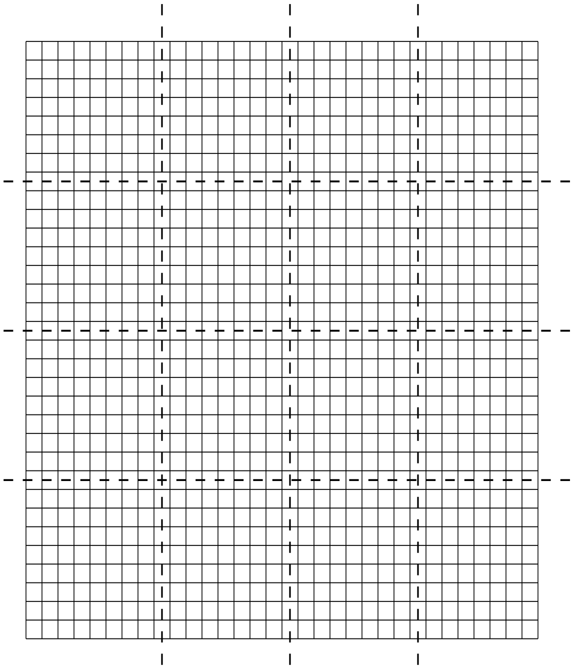

FIG. 2. *Block-structured grid of type used in numerical experiments:* 33×33 *grid with* $M = 8$*.*

does not divide $2^k - 1$, some blocks will be smaller: either $M \times (M-1), (M-1) \times M$, or $(M-1) \times (M-1)$. For example, most of the blocks in the 33×33 grid shown in figure (2) contain 8 interior vertices in each direction, but those on the top boundary include only 7 interior vertices in the vertical direction. Likewise, those on the right boundary include only 7 interior vertices in the horizontal direction.

This block structure only applies in the relaxation process, which is as follows. Beginning at the left boundary, each vertical line is scanned in turn. Within each line the blocks are scanned (beginning at the bottom boundary) and unknowns on the vertical line that are within the block and the first δ unknowns in the adjacent blocks are updated simultaneously to satisfy their discrete equations. The dark dots in figure (3) indicate which unknowns would be updated simultaneously when $M = 8$ and $\delta = 3$. This relaxation scheme can be viewed as a block Gauss-Seidel scheme with overlapping blocks. When $M = 1, \delta = 0$ (i.e. each point is treated as a block) we have pointwise Gauss-Seidel relaxation, and when $M = 2^k + 1$ (i.e. the whole grid is treated as one block) we have y-line Gauss-Seidel relaxation.

We report results for a two-level $V(1,1)$ cycle. The fine grid is 257×257, and the global coarse grid problem (on a 129×129 grid) is solved exactly. Bilinear interpolation is used to interpolate the correction to the fine grid and full weighting is used to restrict residuals to the coarse grid. In the experiments the right-hand side, f, is set to zero and homogeneous Dirichlet boundary conditions are applied. Thus the solution to the PDE, and the resulting discrete equations, is the trivial one: $u = 0$. We use a random initial guess, perform many v-cycles, and monitor the convergence factor per cycle (the ℓ_2 norm of the residual after a cycle divided by the ℓ_2 norm of the

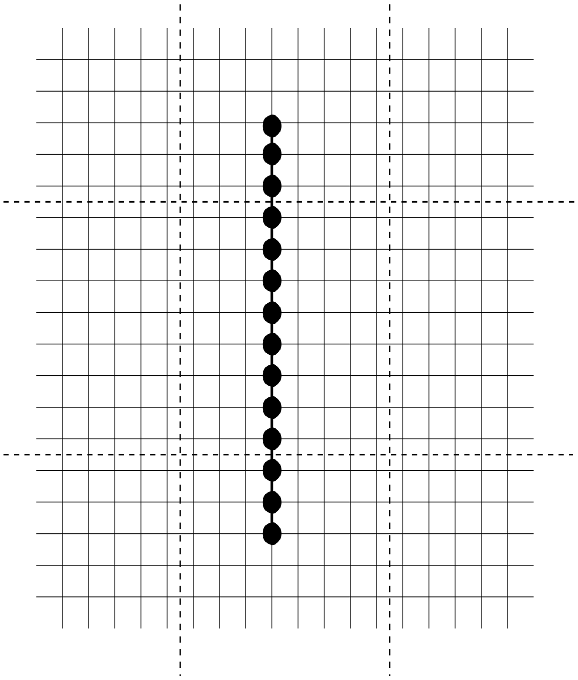

FIG. 3. *Unknowns updated simultaneously: $M = 8$ and $\delta = 3$.*

residual before a cycle). In Table (1) we report the asymptotic convergence factors for various values of the parameters ϵ, M, and δ. Here the asymptotic convergence factor is the worst case (largest) convergence factor observed in 50 cycles. As a point of reference, the asymptotic convergence factor for the isotropic problem, $\epsilon = 1$, using point Gauss-Seidel relaxation, $M = 1, \delta = 0$, is approximately 0.2. If we want to maintain this factor for anisotropic problems by using block-wise line relaxation, it is clear from the table that both the required size of the blocks M and the amount of overlap δ grow with the strength of the anisotropy. The results in the table (in particular, the similar convergence factors for the three entries: $\epsilon = 1/40, M = 4, \delta = 2$; $\epsilon = 1/160, M = 8, \delta = 4$; and $\epsilon = 1/640, M = 16, \delta = 8$) suggest the following:

OBSERVATION 2.1. *To obtain convergence factors similar to those for the isotropic problem, the minimum block size is $\mathcal{O}(1/\sqrt{\epsilon})$. Further, for this minimum block size the required overlap is also $\mathcal{O}(1/\sqrt{\epsilon})$.*

3. Analysis.

Standard local mode analysis of the block-wise line relaxation scheme used in the above numerical experiments is complicated by the fact that a relaxation sweep does not transform a given Fourier component into a simple multiple of itself. Because of this, in the following analysis we will use algebraic arguments to see what combination of parameters ϵ, M, and δ can guarantee that block-wise line relaxation smoothes the error as effectively as full line relaxation.

We consider the same model problem (2D anisotropic diffusion equation discretized by finite differences on a uniform grid of size $n \times n$), but with periodic boundary conditions. Given some initial approximate solution $\mathbf{U}^h$, the result of full line relaxation of the line with i index equal to I can be written as

$$(3) \qquad \mathbf{u}^{line} = L^{-1}(h^2\mathbf{f}_I - \epsilon\mathbf{U}_{I-1} - \epsilon\mathbf{U}_{I+1}).$$

Here

$$(4) \qquad \mathbf{f}_I = \begin{bmatrix} f^h_{I,1} \\ f^h_{I,2} \\ \cdot \\ \cdot \\ \cdot \\ f^h_{I,n} \end{bmatrix}, \mathbf{U}_{I-1} = \begin{bmatrix} U^h_{I-1,1} \\ U^h_{I-1,2} \\ \cdot \\ \cdot \\ \cdot \\ U^h_{I-1,n} \end{bmatrix}, \mathbf{U}_{I+1} = \begin{bmatrix} U^h_{I+1,1} \\ U^h_{I+1,2} \\ \cdot \\ \cdot \\ \cdot \\ U^h_{I+1,n} \end{bmatrix},$$

and the $n \times n$ matrix

$$L = \begin{bmatrix} -2(1+\epsilon) & 1 & 0 & 0 \;\cdot\;\cdot\;\cdot & & & 0 & 1 \\ 1 & -2(1+\epsilon) & 1 & 0 \;\cdot\;\cdot\;\cdot & & & 0 & 0 \\ 0 & 1 & -2(1+\epsilon) & 1 \;\cdot\;\cdot\;\cdot & & & 0 & 0 \\ & & & \vdots & & & & \\ 0 & 0 & \cdot & \cdot\;\cdot & 0 & 1 & -2(1+\epsilon) & 1 \\ 1 & 0 & \cdot & \cdot\;\cdot & 0 & 0 & 1 & -2(1+\epsilon) \end{bmatrix}.$$

Given the same initial approximate solution, the result of relaxing the unknowns with index $(I,j); j = 1,...,N < n$ as a block can be similarly written as

$$(5) \quad \mathbf{u}^{block} = \tilde{L}^{-1}(h^2\mathbf{f}_I[1:N] - \epsilon\mathbf{U}_{I-1}[1:N] - \epsilon\mathbf{U}_{I+1}[1:N] - U^h_{I,n}\mathbf{e}_1 - U^h_{I,N+1}\mathbf{e}_N).$$

Here we have used the notation $\mathbf{f}_I[1:N]$ to denote the N-dimensional vector formed from the first N components of $\mathbf{f}_I$, and similarly $\mathbf{U}_{I-1}[1:N]$ and $\mathbf{U}_{I+1}[1:N]$. The vectors $\mathbf{e}_1$ and $\mathbf{e}_N$ are the N-dimensional unit vectors with a one for the first and N-th component respectively, and the $N \times N$ matrix

$$\tilde{L} = \begin{bmatrix} -2(1+\epsilon) & 1 & 0 & 0 \;\cdot\;\cdot\;\cdot & & & 0 & 0 \\ 1 & -2(1+\epsilon) & 1 & 0 \;\cdot\;\cdot\;\cdot & & & 0 & 0 \\ 0 & 1 & -2(1+\epsilon) & 1 \;\cdot\;\cdot\;\cdot & & & 0 & 0 \\ & & & \vdots & & & & \\ 0 & 0 & \cdot & \cdot\;\cdot & 0 & 1 & -2(1+\epsilon) & 1 \\ 0 & 0 & \cdot & \cdot\;\cdot & 0 & 0 & 1 & -2(1+\epsilon) \end{bmatrix}.$$

Letting $\mathbf{u}^{line}[1:N]$ denote the N-dimensional vector formed from the first N components of $\mathbf{u}^{line}$, this vector satisfies

$$(6)\ \mathbf{u}^{line}[1:N] = \tilde{L}^{-1}(h^2\mathbf{f}_I[1:N] - \epsilon\mathbf{U}_{I-1}[1:N] - \epsilon\mathbf{U}_{I+1}[1:N] - u^l_n\mathbf{e}_1 - u^l_{N+1}\mathbf{e}_N).$$

Defining $\mathbf{d}$ to be the difference between the results of full line relaxation and block-wise line relaxation, i.e. $d_j = u_j^{line} - u_j^{block}, j = 1, ..., N$, yields (by subtracting equation(5) from equation(6))

$$\mathbf{d} = -(u_n^{line} - U_{I,n}^h)\tilde{L}^{-1}\mathbf{e}_1 - (u_{N+1}^{line} - U_{I,N+1}^h)\tilde{L}^{-1}\mathbf{e}_N. \tag{7}$$

In the following, we consider only the second term above; the analysis for the first term is completely analogous. This second term is the product of two factors: $u_{N+1}^{line} - U_{I,N+1}^h$, the change induced at grid point $(I, N+1)$ by line relaxation, and $\tilde{L}^{-1}\mathbf{e}_N$. In the analysis, we would like to get an upper bound on magnitude of $\mathbf{d}$ and see how this bound depends on the parameters ϵ and N. In general, it is not possible to bound the scalar factor $u_{N+1}^{line} - U_{I,N+1}^h$ because changing the value of the initial approximation at the point $(I, N+1)$ will not effect the result of line relaxation. In the following, we make the assumption that this scalar factor is $= 1$. One could carry this factor throughout the analysis, or simply multiply our final bound on $\mathbf{d}$ by this factor, but doing so would only obscure (but not invalidate) the points we are making. With these assumptions the equation relating the difference between full line relaxation and block-wise line relaxation is

$$\mathbf{d} = -\tilde{L}^{-1}\mathbf{e}_N \tag{8}$$

Using Gaussian elimination to solve equation(8), one obtains the following formula for the components of $\mathbf{d}$:

$$d_j = p_{j-1}(\epsilon)/p_N(\epsilon), j = 1, ..., N, \tag{9}$$

where $p_k(\epsilon)$ is a polynomial in ϵ of degree k. The polynomials are generated by the recurrence relation

$$\begin{aligned} p_0(\epsilon) &= 1 \\ p_1(\epsilon) &= 2 + 2\epsilon \\ p_k(\epsilon) &= (2+2\epsilon)p_{k-1}(\epsilon) - p_{k-2}(\epsilon) \quad k = 2, ..., N \end{aligned} \tag{10}$$

Defining $\alpha_k = p_k(\epsilon)/p_{k-1}(\epsilon)$, we can rewrite the recurrence relation in the form

$$\begin{aligned} \alpha_1 &= 2 + 2\epsilon \\ \alpha_k &= 2 + 2\epsilon - 1/\alpha_{k-1} \quad k = 2, ..., N \end{aligned} \tag{11}$$

Theorem 3.1. *For the sequence defined by the above recurrence relation*

$$\alpha_k \geq 1 + \epsilon + \sqrt{2\epsilon + \epsilon^2}, \forall k$$

Proof: Define

$$g(x) = 2 + 2\epsilon - 1/x.$$

Then $g \in C[1+2\epsilon, 2+2\epsilon]$ *and* $g(x) \in [1+2\epsilon, 2+2\epsilon], \forall x \in [1+2\epsilon, 2+2\epsilon]$. *Further,*

$$0 < g'(x) \leq 1/(1+2\epsilon)^2 < 1, \text{ for } x \in (1+2\epsilon, 2+2\epsilon).$$

These conditions guarantee convergence of the sequence $\{\alpha_k\}_1^\infty$ to the fixed point $\alpha = 1+\epsilon+\sqrt{2\epsilon+\epsilon^2}$ (See, for example, the fixed point theorems in Chapter 2 of [3]). In addition, by application of the mean value theorem

(12)
$$\begin{aligned}\alpha_k - \alpha &= g(\alpha_{k-1}) - g(\alpha)\\ &= g'(\zeta)(\alpha_{k-1}-\alpha), \zeta \in (1+2\epsilon, 2+2\epsilon).\end{aligned}$$

Because $0 < g'(x) < 1$ for $x \in (1+2\epsilon, 2+2\epsilon)$ and $\alpha_1 > \alpha$, equation(12) implies that the sequence is monotonically decreasing. The theorem follows from the convergence and the monotonicity.

From this theorem, we have the following bound on $\mathbf{d}$, the difference between the results for line relaxation and block-wise line relaxation:

(13)
$$\begin{aligned}|d_j| &= p_{j-1}(\epsilon)/p_N(\epsilon)\\ &= \prod_{k=j}^N p_{k-1}(\epsilon)/p_k(\epsilon)\\ &= \prod_{k=j}^N (\alpha_k)^{-1}\\ &\leq (1+\epsilon+\sqrt{2\epsilon+\epsilon^2})^{-(N-j+1)}.\end{aligned}$$

Given some tolerance $\gamma < 1$, we can guarantee that $|d_j|$ will be less than γ by requiring that

(14)
$$j \leq N + \frac{\log(\gamma)}{\log(1+\epsilon+\sqrt{2\epsilon+\epsilon^2})}.$$

This equation can be viewed as a requirement on N, it must be at least large enough so that the right-hand side of the inequality is positive. For a fixed ϵ, if we choose

(15)
$$N = N_\epsilon = \left(\frac{1}{\beta-1}\right)\frac{\log(\gamma)}{\log(1+\epsilon+\sqrt{2\epsilon+\epsilon^2})}, 0 < \beta < 1,$$

then $|d_j| < \gamma$ for $j = 1, 2, \ldots, \beta N$. With this choice of N the difference between line relaxation and block-wise line relaxation will be less than the tolerance except at points near the end of the block - those with index $(I, j), \beta N < j \leq N$. If ϵ is now reduced by the factor $0 < \mu < 1$, to meet the same criterion ($|d_j| < \gamma$ for $j = 1, 2, \ldots, \beta N$) we need

(16)
$$N = N_{\mu\epsilon} = \left(\frac{1}{\beta-1}\right)\frac{\log(\gamma)}{\log(1+\mu\epsilon+\sqrt{2\mu\epsilon+\mu^2\epsilon^2})}.$$

It follows from equations(15) and (16) that

(17)
$$N_{\mu\epsilon} = N_\epsilon \frac{\log(1+\epsilon+\sqrt{2\epsilon+\epsilon^2})}{\log(1+\mu\epsilon+\sqrt{2\mu\epsilon+\mu^2\epsilon^2})}.$$

And in the limit as $\epsilon \longrightarrow 0$,

(18)
$$N_{\mu\epsilon} = N_\epsilon/\sqrt{\mu}.$$

The results of this analysis agree with Observation 2.1 in the numerical experiments section. Namely, equation(14) shows that relaxing only a segment of line can smooth as effectively as full line relaxation at points away from the end points of the segment provided that N, the length of the segment, is large enough. However, it may not be an effective smoother near the endpoints. Therefore, to get effective smoothing within a block using block-wise line relaxation we must extend the line segments relaxed into the neighboring blocks. Further, equation(18) shows that for small ϵ the length of the line segment must be $\mathcal{O}(1/\sqrt{\epsilon})$ to guarantee effective smoothing away from the endpoints, i.e. for $(j \leq \beta N)$. This means that block-wise line relaxation can be an effective smoother provided that both the block size M and the overlap δ are $\mathcal{O}(1/\sqrt{\epsilon})$.

4. Conclusions. The result of relating the necessary block size and overlap in a multi-block line relaxation scheme to the strength of the anisotropy that was illustrated by the numerical results in Section 2, appears to hold for many schemes. Some of these may be more efficient on parallel computers than the scheme in Section 2.

For example, damped vertical line Jacobi relaxation (with damping parameter $\omega = 0.7$) is known to be an effective smoother for our model problem (see [4]). Figure (4) illustrates a multi-block version of this relaxation scheme (with block size $M = 6$ and overlap $\delta = 1$). Given an initial approximation $\mathbf{U}$, in step 1 a new approximation $\mathbf{V}$ is calculated at the indicated points by block relaxation of each of the marked line segments using values of $\mathbf{U}$ at all points not on that line segment. In step 2, a new approximation $\mathbf{V}$ is again calculated at the indicated points by by block relaxation of each of the marked line segments using values of $\mathbf{U}$ at all points not on that line but the value of $\mathbf{V}$ calculated in step 1 for points directly above and below the line segment. All line segments in step 1 can be processed simultaneously, likewise in step 2. Then the final update is given by

$$\mathbf{U} \leftarrow (1-\omega)\mathbf{U} + \omega\mathbf{V},\ \omega = 0.7.$$

Note that the scheme is simply damped vertical line Jacobi when the grid is treated as a single block. We have experimented with this scheme, and have found that the resulting multigrid algorithm is efficient (convergence factors are as good or better than those for damped Jacobi on the isotropic problem) provided that the block size and overlap are $\mathcal{O}(1/\sqrt{\epsilon})$.

For simplicity, our analysis and test problems have focused on rectangular grids. However, we would expect the same block size and overlap requirements would need to be met for an efficient multigrid solver on a more general block-structured grid - such as the one in figure(1). A practical algorithm would likely involve checking to see if there is an anisotropy that crosses the block interface and, if so, constructing a sensible extension of the line from one block to its neighbor. This is an easier process to implement than constructing a "global" line for such a grid. We are currently investigating this idea (and 3D generalizations) with the aim of developing an efficient relaxation process to be used in multigrid solvers for general, three dimensional block-structured grids.

5. Acknowledgements. The authors would like to acknowledge their indebtedness to Achi Brandt for the idea of this paper. Brief mention of the observation that

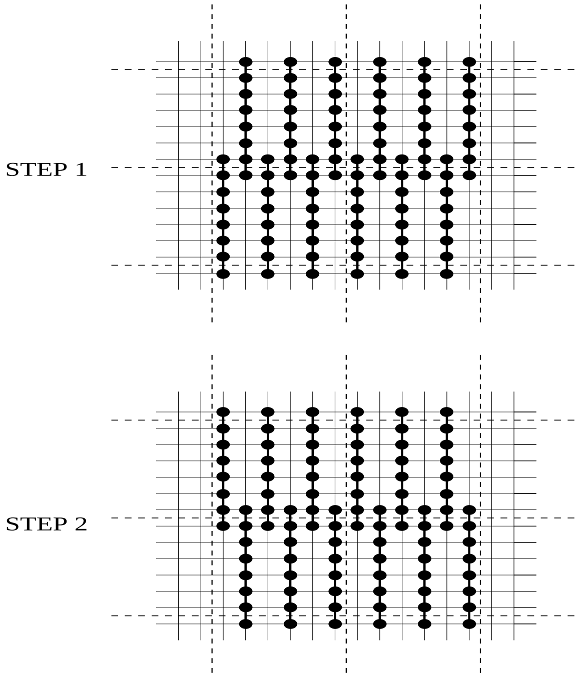

FIG. 4. *Multi-block relaxation based on damped line Jacobi*

the block size must be $\mathcal{O}(1/\sqrt{\epsilon})$ can be found in [2]. The observation that the overlap must be $\mathcal{O}(1/\sqrt{\epsilon})$ was also communicated to the authors by Brandt. To the best of the authors' knowledge, this paper contains the first published numerical results and analysis.

REFERENCES

[1] A. Brandt. Multi–level adaptive solutions to boundary–value problems. *Math. Comp.*, 31:333–390, 1977.

[2] A. Brandt. Multigrid solvers on parallel computers. In M. Schultz, editor, *Elliptic Problem Solvers*, pages 39 – 84. Academic Press, 1981.

[3] R.L. Burden and J.D. Faires. *Numerical Analysis, third edition.* Prindle, Weber, and Schmidt, 1985.

[4] P. Wessling. *An Introduction to Multigrid Methods.* John Wiley and Sons, 1992.

$\epsilon = 1/10$						
δ	$M = 1$	$M = 2$	$M = 4$	$M = 8$	$M = 16$	$M = 32$
0	.690	.498	.305	.196	.170	.166
1		.108	.108			
$\epsilon = 1/40$						
δ	$M = 1$	$M = 2$	$M = 4$	$M = 8$	$M = 16$	$M = 32$
0	.898	.815	.683	.530	.432	.412
1		.493	.368	.181	.083	.093
2			.177			
$\epsilon = 1/160$						
δ	$M = 1$	$M = 2$	$M = 4$	$M = 8$	$M = 16$	$M = 32$
0	.962	.939	.891	.812	.719	.648
1		.814	.745	.573	.393	.288
2			.573	.406	.220	.125
3				.317	.129	
4				.216		
$\epsilon = 1/640$						
δ	$M = 1$	$M = 2$	$M = 4$	$M = 8$	$M = 16$	$M = 32$
0	.978	.972	.958	.935	.893	.836
1		.938	.918	.851	.739	.604
2			.852	.774	.614	.441
3				.698	.511	.324
4				.606	.427	.239
5					.360	.178
6					.306	
7					.261	
8					.203	
$\epsilon = 1/2560$						
δ	$M = 1$	$M = 2$	$M = 4$	$M = 8$	$M = 16$	$M = 32$
0	.978	.977	.975	.970	.953	.925
1		.971	.966	.948	.905	.834
2			.947	.924	.861	.757
3				.891	.809	.670
4				.854	.759	.605
5					.723	.545
6					.685	.492
7					.644	.442
8					.590	.395
9						.359
10						.320
11						.289
12						.259
13						.231
14						.210
15						.189
16						.160

Table 1

Asymptotic multigrid convergence factors