NUMERICAL ANALYSIS PROJECT
MANUSCRIPT *NA-92-14*

SEPTEMBER 1992

Cyclic Reduction/Multigrid

by

Gene H. Golub
and
Ray S. Tuminaro

NUMERICAL ANALYSIS PROJECT
COMPUTER SCIENCE DEPARTMENT
STANFORD UNIVERSITY
STANFORD, CALIFORNIA 94305

CYCLIC REDUCTION/MULTIGRID

GENE H. GOLUB* AND RAY S. TUMINARO†

August 5, 1992

Abstract. We consider the use of the multigrid method in conjunction with a cyclic reduction preconditioner for convection-diffusion equations. This preconditioner corresponds to algebraically eliminating all the unknowns associated with the red points on a standard mesh colored in a checkerboard fashion. It is shown that the multigrid method applied to the resulting operator often converges much faster than when applied to the original equations. Fourier analysis of a constant coefficient model problem as well as numerical results for nonconstant coefficient examples are used to validate the conclusions.

1. Introduction. A new method is presented for the solution of the convection-diffusion equation

$$-\nabla \cdot p(x,y)\nabla u + q(x,y)\cdot \nabla u = f \qquad \text{on } \Omega \tag{1}$$

with

$$r(x,y)u + s(x,y)u_n = g \qquad \text{on } \partial\Omega.$$

This equation is of fundamental importance in a number of scientific areas (e.g. computational fluid dynamics). However, using iterative methods to obtain solutions of the corresponding discrete equations is often problematic. Many standard iterative methods converge slowly (or even diverge) when applied to discrete systems corresponding to highly convective or unstable behavior.

In this paper, we consider a multigrid solution technique in conjunction with one step of cyclic reduction. Specifically when a five-point discretization is used on a uniform grid, the cyclic reduction procedure corresponds to first coloring the grid in

* Computer Science Dept., Stanford University, Stanford, Ca. This author was supported by the National Science Foundation under grant DCR-8412314.

† Parallel Algorithm Project, CERFACS, Toulouse, France. Part of this work was performed at Sandia National Laboratories and supported by the Applied Mathematical Science Program, U.S. Department of Energy, Office of Energy Research.

a checkerboard fashion and then algebraically eliminating all the unknowns associated with the red points. The resulting set of equations correspond to a nine-point discretization of a new partial differential equation. The multigrid technique is now applied to this new system of equations. The key point is that the new system is half the size of the original and that this system is often easier to solve with iterative methods than the original.

The use of cyclic reduction as a **preconditioner** has been studied in several papers within other iterative solvers and for other types of equations. In [1], Axelsson and Gustafsson report improved convergence rates using cyclic reduction with the conjugate gradient method for symmetric systems. Elman and Golub ([5], [6], and [7]) illustrated and analyzed the advantages of using cyclic reduction for nonsymmetric convection-diffusion operators in conjunction with a number of iterative and block iterative solvers. Simpson [13] has reported convergence benefits using cyclic reduction for dynamic, two species, reaction-diffusion systems of partial differential equations. In this paper, we show that similar results are possible with the multigrid scheme. That is, the multigrid method often converges significantly faster when applied to the algebraically reformulated equations than to the original. Further, there are instances where the multigrid method diverges if applied to the original system, but converges quite satisfactorily for the reduced system. A Fourier analysis is used to illustrate this property. Additionally, numerical results are given for a wide range of PDE problems.

In Section 2 we discuss the convection-diffusion equation and its solution by traditional multigrid methods. This is followed by a description of the cyclic reduction process in Section 3 . In Section 4, cyclic **reduction/multigrid** is developed. Special **attention** is given to the resulting system of equations and the corresponding grid. Smoothing analysis of the **Gauss-Seidel** operator on the resulting reduced equations is pursued in Section 5. Finally, the **paper** concludes with experimental results in Section 6 and general observations in Section 7.

2. Multigrid **methods and the convection/diffusion equation.** The multigrid algorithm is a fast and efficient method for solving the systems of equations that arise from many PDE applications. We give only a brief sketch of a typical multigrid algorithm. Detailed descriptions of more general multigrid algorithms can be found in [2] and [9]. Excellent introductions to multigrid can be found in [4] and [10].

```
/* Solve the equations A_level u = b */
Procedure MG(b, u, level)
    if (level == Coarsest ) then u ← A_level^{-1} b
    else
        u ←relax( b, u, level)
        r ← b − Au
        r̃ ← Rr          /* R is a projection opemtor */
        v ← 0
        v ← MG(r̃, v, level + 1)
        u ← u + Pv   /* P is an interpolation opemtor */
    endif
```

FIG. 1. *MG Algorithm*

One iteration of a simple multigrid 'V' cycle consists of smoothing the error using a relaxation technique, 'solving' an approximation to the smooth error equation on a coarse grid, interpolating the error correction to the fine grid, and finally adding the error correction into the approximation. An important aspect of the multigrid method is that the coarse grid solution can be approximated by recursively using the multigrid idea. That is, on the coarse grid, relaxation is performed to reduce high frequency errors followed by the projection of a correction equation on yet a coarser grid, and so on. Thus, the MG method requires a series of problems to be 'solved' on *a* hierarchy of grids with different mesh sizes. We summarize one iteration of this procedure in Figure 1.

Application of the multigrid algorithm to the convection-diffusion equation requires some care. To understand many of the potential difficulties, we consider the constant coefficient convection-diffusion equation

$$-\Delta u + \sigma u_x + \tau u_y = f \tag{2}$$

on the unit square with Dirichlet boundary conditions. The system is discretized via central difference and yields the following computational stencil at the interior grid

points:

$$\frac{1}{h^2}\begin{pmatrix} & -1+\delta & \\ -1+\gamma & 4 & -1-\gamma \\ & -1-\delta & \end{pmatrix} \tag{3}$$

where $\delta = \frac{\tau h}{2}$, $\gamma = \frac{\sigma h}{2}$ and h is the mesh spacing.

Unfortunately, the application of the multigrid method to (3) becomes more problematic as δ and 7 become larger. In particular, the discrete difference operator associated with (3) is no longer diagonally dominate for large δ and 7 and so most classical iterative methods (e.g. Gauss-Seidel, Jacobi, etc.) diverge when applied to this problem. For multigrid methods, the above difficulties with classical iterative schemes correspond to difficulties in finding a relaxation procedure that properly smoothes high frequencies. Notice that even if 6 and 7 are fairly modest on the finest grid, it may still be difficult to find an appropriate smoother on the coarse levels due to the fact that δ and 7 become larger on coarser grids. Thus, a simple multigrid method applied to (3) will most likely diverge when δ and 7 are large due to the divergence of the smoother.

To reduce the problems associated with the convection diffusion-operator, a number of multigrid approaches have been considered (see [3], [9] and [11]). For example, the use of downstream relaxation can alleviate many difficulties for moderate 6 and 7 (see [3]). That is, when the points are relaxed in the direction of the flow with upstream points begin relaxed first, the smoothing properties of the relaxation scheme are usually enhanced. Another possibility is to add artificial dissipation' to the discrete operator. In particular, the convergence difficulties associated with large 6 and 7 can be viewed in terms of t ' stability of the discrete approximation. That is, as δ and 7 become larger, the discretization becomes increasingly unstable making the successful application of most classical iterative methods unlikely. In fact, in the limit as δ and 7 approach infinity the equations at the even and odd points completely decouple resulting in large oscillations in the solution. To rectify the situation, artificial dissipation can be introduced to generate equations which are more stable and thereby more amenable to standard smoothers. Though this may be somewhat unsatisfactory

[1] The PDE that one wishes to solve is modified by adding terms containing second (and possibly fourth) derivatives. This effectively adds more diffusion into the problem.

on the finest grid since a different set of equations are being solved, it is quite suitable on coarse meshes.

3. Cyclic reduction. We now describe the cyclic reduction process for transforming the original equations into a more well-behaved system.

Consider the discrete system

$$Au = b.$$

When the matrix A corresponds to a five-point finite difference discretization, it has property-A [14]. That is, its rows and columns can be symmetrically permuted to the form

$$\begin{pmatrix} D & C \\ E & F \end{pmatrix} \begin{pmatrix} u^{(r)} \\ u^{(b)} \end{pmatrix} = \begin{pmatrix} f^{(r)} \\ f^{(b)} \end{pmatrix} \tag{4}$$

where D and F are diagonal matrices and (4) corresponds to a red-black ordering of the underlying grid. By eliminating the unknowns $u^{(r)}$, we can produce the reduced system

$$[F - ED^{-1}C]u^{(b)} = f^{(b)} - ED^{-1}f^{(r)}.$$

The new system of equations corresponds to a 9 point finite difference operator (for the interior grid points). We omit the details (see [5]) and simply state that one cyclic reduction step applied to the matrix associated with (3) yields an operator with the following stencil at the interior grid points:

$$\begin{pmatrix} & (6-1)^2 & \\ 2(1-\gamma)(1-6) & & 2(1+\gamma)(1-\delta) \\ (\gamma-1)^2 & -16 + 2(1-\delta^2) + 2(1-\gamma^2) & (1+\gamma)^2 \\ 2(1-\gamma)(1+\delta) & & 2(1+\gamma)(1+\delta) \\ & (1+\delta)^2 & \end{pmatrix}. \tag{5}$$

Similar operators can be worked out for the boundary stencils.

We conclude this section by describing a few of the connections between cyclic reduction and other approaches. For example, the one step cyclic reduction procedure can be viewed as a specific type of domain decomposition method (see [8] for discussions of other types of domain decomposition) where each domain contains only

one interior grid point. In this case the reduced equations correspond to the interface equations which have been explicitly computed. While the reduced operator has been derived by purely algebraic means, it is also possible to generate the same equations by considering a slightly different PDE. In particular, the original convection-diffusion equation (2) can be differentiated multiple times with respect to x and y. Taking appropriate linear combinations, the following equivalent PDE is obtained:

$$
\begin{aligned}
&(1+\frac{h^2\sigma^2}{8})u_{xx}+(1+\frac{h^2\tau^2}{8})u_{yy}+\sigma u_x+\tau u_y+\frac{\sigma\tau h^2}{4}u_{xy}+\\
&\frac{h^2}{4}(\sigma u_{yyx}+\tau u_{xxy}+\sigma u_{xxx}+\tau u_{yyy})+\\
&\frac{h^2}{8}(u_{xxxx}+2u_{xxyy}+u_{yyyy}) \;=\; f+\frac{h^2}{8}(\sigma f_x+\tau f_y+f_{xx}+f_{yy})
\end{aligned}
\tag{6}
$$

where h is some constant which will be taken as the mesh spacing for the discrete equations. If the first and second order terms in (6) are discretized in a specific fashion (see appendix), the reduced stencil (5) is again obtained. This implies that one cyclic reduction step is equivalent to the introduction of the terms $(h^2\sigma^2/8)u_{xx}$, $(h^2\tau^2/8)u_{yy}$ and $\frac{\sigma\tau h^2}{4}u_{xy}$ into the discretization making the discrete equations more diffusive in nature. Thus, the cyclic reduction approach is quite similar to the addition of artificial dissipation. However, the key point to recognize about cyclic reduction is that it does not change the solution to the underlying equations which are being solved (as artificial dissipation does). The solution we obtain on the black points is the solution to the original PDE at the black points and the solution at the red points can be easily recovered using *(4).*

4. Cyclic reduction/multigrid. The cyclic reduction multigrid algorithm consists of first applying one step of cyclic reduction to the discrete set of equations and then applying the multigrid algorithm to the resulting reduced system. In this section we discuss some multigrid issues that must be resolved.

To apply the multigrid method, a grid hierarchy must be constructed: $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3, \ldots$ where $\mathcal{G}_1$ corresponds to the finest grid, $\mathcal{G}_2$ corresponds to the next finest grid, etc. Given that the reduced stencil is defined on a somewhat irregular grid (corresponding to the black points in a checkerboard), some thought must be given to appropriate grid hierarchy. For example, the next coarsest grid can be an evenly spaced mesh with grid spacing $2h$ (where h is the mesh width of the original problem on the fine

grid). That is, half the black points are removed to define the next coarsest grid. This corresponds to using the same coarse grid that would be used if the equations were not cyclically reduced. The advantage of this approach is that the algorithms and coding are similar to the standard (unreduced) case. Another possibility is to consider the entire stencil skewed. That is, we can reinterpret the reduced operator as a standard nine point discretization defined on a diamond shaped grid (using a mesh spacing of $\sqrt{2}h$). The next coarsest grid can then be defined by simply doubling this mesh spacing. This has the advantage that all the grids can be treated in a uniform way. Figure 2 shows the grid hierarchy for the two approaches using three levels.

In our multigrid implementation we have chosen the first strategy for the grid hierarchy. This choice was primarily motivated by programming considerations. We use standard projection and interpolation operators. To project from the finest grid, $\mathcal{G}_1$, to the next grid, $\mathcal{G}_2$, an operator with the stencil

$$\frac{1}{4\sqrt{2}} \begin{pmatrix} 1 & & 1 \\ & & \\ 1 & 4 & 1 \end{pmatrix}$$

is used. To project between other coarse grids, an operator with the stencil

$$\frac{1}{4} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \tag{7}$$

is used. The interpolation operators are the transpose of the projection operators. For a smoothing procedure, Gauss-Seidel is used on each level. To complete the specification of the method, we must define coarse grid difference operators. The general idea is that the coarse grid difference operator should approximate the behavior of the reduced finite difference operator on smooth components. For the grids $\mathcal{G}_i$ and $i > 2$, we use a combination of Galerkin coarsening and artificial dissipation. For A_{i+1}, we first define

$$\tilde{A}_{i+1} = R_i A_i P_i$$

where P_i and R_i are the interpolation and projection operators associated with the i^{th} grid. If $\tilde{A}_{i+1}$ is diagonally dominate, we take

$$A_{i+1} = \tilde{A}_{i+1}.$$

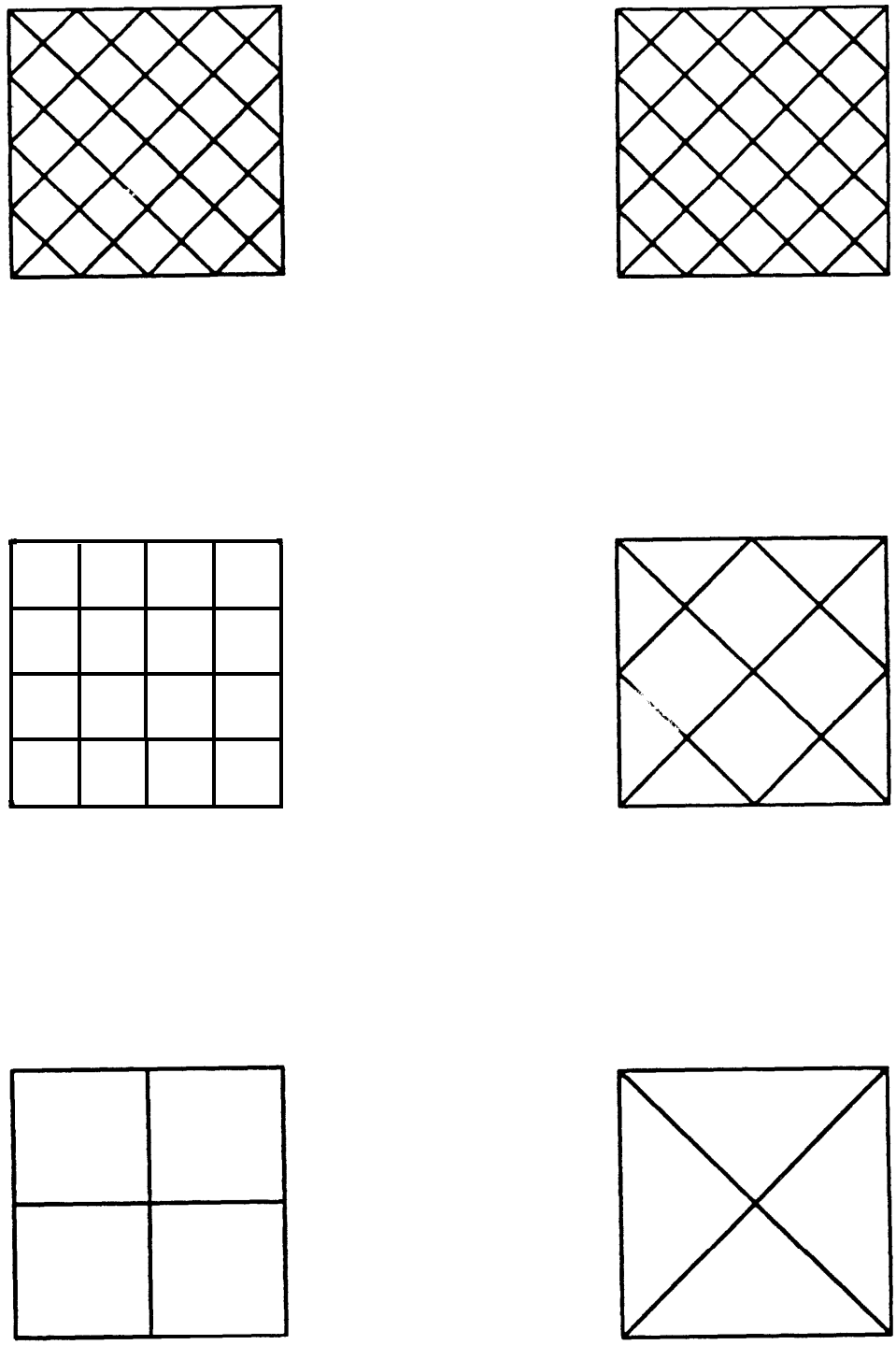

FIG. 2. *Two different choices for the grid hierarchy within a 3 grid method.*

Otherwise, artificial dissipation (second order terms $\epsilon_1 u_{xx}$ and $\epsilon_2 u_{yy}$) is added until the operator is diagonally dominate. It is important to note, however, that no artificial dissipation is added to the finest grid. For the second coarsest grid, $\mathcal{G}_2$, we proceed in a slightly different fashion to avoid growth in the number of elements in the stencil. The operator A_1 is given by the local stencil

$$\begin{pmatrix} & a & \\ b & & c \\ d & e & f \\ & g & h \\ & i & \end{pmatrix}.$$

We split the operator A_1 into 2 components

$$A_1 = A_1^c + A_1^s$$

where A_1^c corresponds to the derivative terms given by the coefficients b, c,g, and h. We use Galerkin projection on A_1^c followed by artificial dissipation to create A_2^c. The coarse grid operator is now given by

$$A_2 = A_2^c + A_2^s.$$

Finally, for purposes of comparison we also apply the multigrid method to the original set of equations using bilinear interpolation and full weighted restriction (7) to transfer between grids, and using the same scheme of Galerkin and artificial dissipation to produce the coarse grid equations.

5. Fourier Smoothing Analysis. We estimate the convergence of the multigrid method using Fourier analysis [2]. Specifically, we use Fourier analysis to determine how well the relaxation procedure damps the high frequency error components.

We illustrate the Fourier smoothing analysis by considering the lexicographical Gauss-Seidel operator applied to a constant coefficient problem with the following stencil

$$\begin{pmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{pmatrix}. \tag{8}$$

For the analysis that follows we assume periodic boundary conditions. Typically, when the equations are stable with a modest amount of dissipation, the periodic boundary

analysis gives accurate convergence predictions for problems with Dirichlet boundary conditions. However, for problems where the equations are slightly unstable this analysis may be inaccurate. If we denote by $e_{k,j}^{(c)}$ the error associated with the approximate solution at the point (C, j) after c Gauss-Seidel iterations, then the following relationship holds

$$
\begin{aligned}
&s_{11}e_{k-1,j-1}^{(c)} + s_{12}e_{k,j-1}^{(c)} + s_{13}e_{k+1,j-1}^{(c)} + \\
&s_{21}e_{k-1,j}^{(c)} + s_{22}e_{k,j}^{(c)} + s_{23}e_{k+1,j}^{(c-1)} + \\
&s_{31}e_{k-1,j+1}^{(c-1)} + s_{32}e_{k,j+1}^{(c-1)} + s_{33}e_{k+1,j+1}^{(c-1)} = 0
\end{aligned} \tag{9}
$$

where it is assumed that the points are relaxed in lexicographic order. The general idea behind smoothing analysis is to determine how well the relaxation procedure damps the high frequencies errors. If the coarse grid correction adequately damps the low frequencies, then the smoothing analysis will accurately predict the asymptotic multigrid convergence rate. To do the analysis, consider a Fourier mode

$$
e_{k,j}^{(c)} = A_{\theta_1,\theta_2}^{(c)} e^{i(\theta_1 k + \theta_2 j)} \qquad |\theta_1|, |\theta_2| \leq \pi
$$

where $i = \sqrt{-1}$, $A_{\theta_1,\theta_2}^{(c)}$ is the amplitude, and $(\theta_1, \theta_2)$are the frequencies in the x and y directions. One simple definition of the smoothing number is

$$
\mu = \max_{max(\theta_1,\theta_2) \geq \frac{\pi}{2}} \frac{|A_{\theta_1,\theta_2}^{(c)}|}{|A_{\theta_1,\theta_2}^{(c-1)}|}.
$$

That is, the smoothing number is the damping factor associated with the high frequency mode that is damped the least. Using (9) this can be rewritten as

$$
\mu = \max_{max(\theta_1,\theta_2) \geq \frac{\pi}{2}} \left| \frac{s_{23}e^{i\theta_1} + s_{31}e^{i(-\theta_1+\theta_2)} + s_{32}e^{i\theta_2} + s_{33}e^{i(\theta_1+\theta_2)}}{s_{11}e^{i(-\theta_1-\theta_2)} + s_{12}e^{-i\theta_2} + s_{13}e^{i(\theta_1-\theta_2)} + s_{21}e^{-i\theta_1} + s_{22}} \right|. \tag{10}
$$

Thus, to compare standard multigrid with cyclic reduction/multigrid we must compute the smoothing numbers for the operators (3) and (5). Substituting (3) and (5) into (10) we get the following smoothing numbers:

$$
\mu(S) = \max_{max(\theta_1,\theta_2) \geq \frac{\pi}{2}} \left| \frac{(\mathrm{y}+1)e^{i\theta_1} + (\delta+1)e^{i\theta_2}}{(6-1)e^{-i\theta_2} + (\mathrm{y}-1)e^{-i\theta_1} + 4} \right|
$$

and

$$
\mu(R) = \max_{max(\theta_1,\theta_2) \geq \frac{\pi}{2}} \left| \frac{\begin{array}{c} 2(1+\mathrm{y})(1-\delta)e^{i\theta_1} + 2(\delta-1)(\mathrm{y}-1)e^{i\theta_2} \\ + (1-\gamma)^2 e^{i(-\theta_1+\theta_2)} + (1-\delta)^2 e^{i(\theta_1+\theta_2)} \end{array}}{\begin{array}{r} (1+\delta)^2 e^{-i(\theta_1+\theta_2)} + (1+\gamma)^2 e^{i(\theta_1-\theta_2)} - 12 - 2(\delta^2+\gamma^2) \\ + 2(1-\gamma)(1+\delta)e^{-i\theta_1} + 2(1+\gamma)(1+\delta)e^{-i\theta_2} \end{array}} \right|
$$

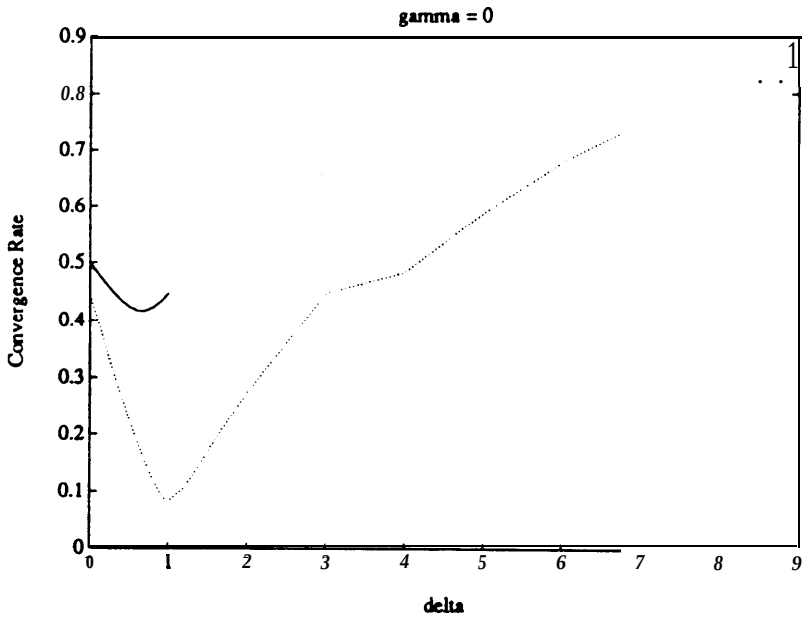

FIG. 3. *Smoothing numbers for lexicographic Gauss-Seidel when $\gamma = 0$. Lower curve corresponds to the reduced opemtor while the upper denotes the original operator.*

where $\mu(S)$ and $\mu(R)$ denote the smoothing factors for the standard and reduced stencils respectively.

Finding the maximum of the above functions analytically is a bit cumbersome, but can often be done over certain ranges of δ and y. Since the expressions are quite involved we instead plot the smoothing numbers as a function of δ and y. Figure 3 illustrates the smoothing numbers for the case y = 0. Figure 4 illustrates the smoothing numbers when y = σ. Figure 5 and Figure 6 shows the smoothing numbers (see appendix for details) when a multicolor Gauss-Seidel is used.[2] In the figures we only plot the convergence estimates for the original (unreduced) system for $\delta \leq 1.0$ as the convergence rates predicted by this analysis (i.e. using periodic boundary conditions) are not accurate for 6 > 1.0 due to the instability of the equations in this regime. For the most part, the two cases given represent extremes in terms of performance. That is, the reduced operator offers the largest performance improvement over the standard operator when y = 0 and the least performance improvement when δ = y. Overall, it is quite clear that there is a noticeable improvement in the convergence rate when the

[2] Note that for a 5 point operator we use standard red-black Gauss-Seidel. However, for the 9 point operators we use a 4 color Gauss-Seidel operator so that no two neighboring points have the same color.

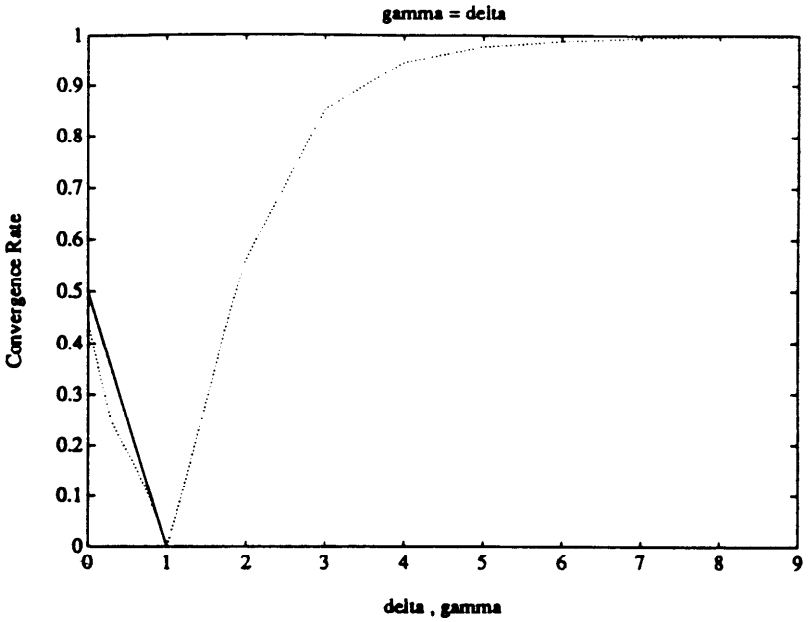

FIG. 4. *Smoothing numbers for lexicographic Gauss-Seidel when* $\gamma = 6$. *Lower curve corresponds to the reduced opemtor while the upper denotes the original operator.*

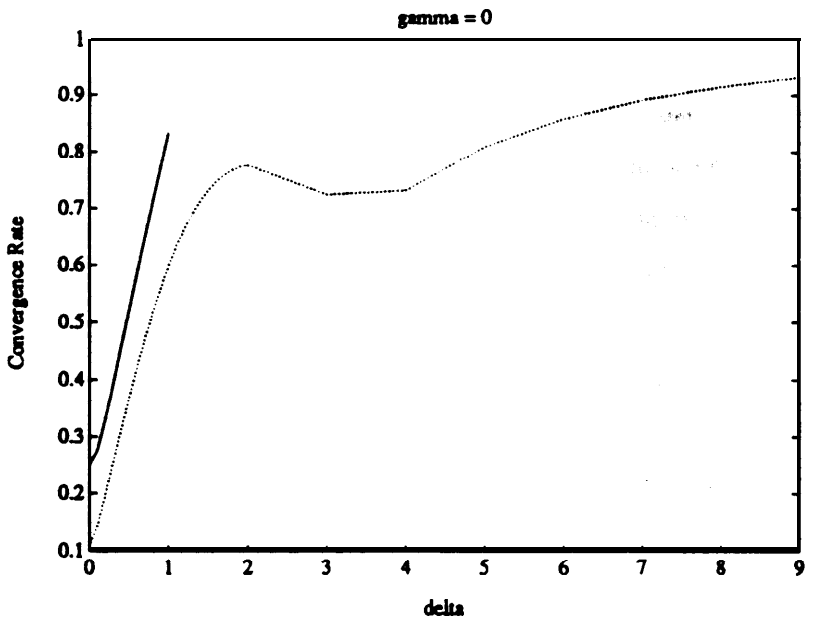

FIG. 5. *Smoothing numbers for multicolor Gauss-Seidel when* $\gamma = 0$. *Lower curve corresponds to the reduced opemtor while the upper denotes the* original *opemtor.*

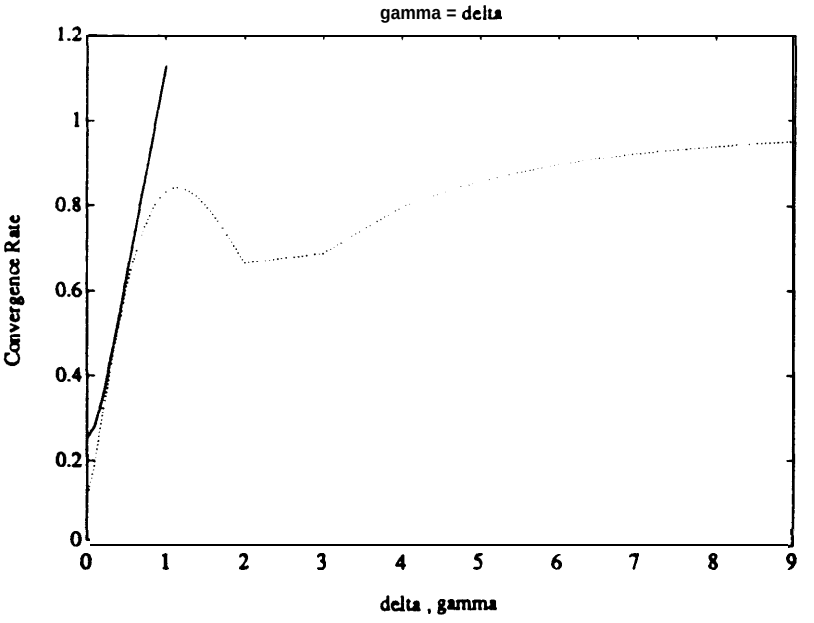

FIG. 6. *Smoothing numbers for multicolor Gauss-Seidel when* $\gamma = \delta$. *Lower curve corresponds to the reduced operator while the upper denotes the original operator.*

reduced operator is used. In many cases the convergence rate is greater than a factor of 2. That is, when the PDE problem is not strongly convective, the smoothing rates for both schemes are satisfactory. However, those associated with the reduced system are significantly better over most ranges of the parameter space. Given that the smoothing rate usually approximates the convergence rate and that the cost per iteration of the two schemes is roughly equivalent (the reduced system requires slightly less work per iteration), we can expect that the computer time required to solve the reduced system will be significantly less than to solve the original system. More importantly, the reduced operator always yields convergent smoothing rates (i.e. $\mu < 1$) as opposed to the the original operator (see Section 6). That is, even when the operator is highly convective or unstable, the corresponding smoothing rates are less than 1. However, it should be noted that the smoothing rates deteriorate (i.e. ≈ 1) for very large δ and y. This is a consequence of the reduced operator becoming increasingly anisotropic.

For example, when $y = 0$ and $\delta \to \infty$ the dominate terms in the operator are

$$\delta^2 \begin{pmatrix} & 1 & \\ 0 & & 0 \\ & 1 & \end{pmatrix}. \tag{11}$$

For anisotropic problems pointwise relaxation is not effective and some form of line-relaxation or semicoarsening should be used in conjunction with multigrid. We do not pursue the subject further and refer the reader to [3].

We conclude this section with heuristic arguments to explain the convergence improvement associated with the algebraic elimination process. As stated earlier, one benefit of cyclic reduction is that it introduces more dissipation into the corresponding PDE operator. For large δ and τ, the added dissipation effectively stabilizes the equations and plays the key role in improving the convergence. However, for small δ and τ the added dissipation is not particularly significant. That is, while the PDE is more diffusive in nature, it is defined on a coarser grid (the mesh spacing is $\sqrt{2}$ times larger than for the original operator) and so locally it may appear more convective (i.e. the $O(1/h)$ convective terms are larger relative to the $O(1/h^2)$ diffusive terms on the coarser mesh). Of course the use of a coarser grid improves the convergence of many iterative methods. However, the convergence of the multigrid method does not depend on the mesh spacing. Instead, the better smoothing rate for small δ and τ is a result of the stencil being spread smoothly over more points making it easier to smooth the error by relaxation. For example, the smoothing properties of the 9 point discrete Laplace operator are better than the smoothing behavior of the standard 5 point Laplace operator (which corresponds to the cyclically reduced 5 point operator). Thus in the case of small δ and γ the benefit is essentially the smooth spreading of the stencil.

8. Experimental Results. A multigrid W-cycle program has been implemented to test the cyclic reduction approach.[3] A number of problems have been tested. In this section we describe the results of this experimentation.

[3] The W cycle differs from the V cycle in the number of times coarse grids are visited. By replicating the line $v \leftarrow MG(\ldots)$, the program fragment given in Fig. 1 can be modified to a W cycle.

Y	original				reduced			
	DS		AS		DS		AS	
	iter	time	iter	time	iter	time	iter	time
0.	23	19.5	23	19.5	14	12.3	14	12.3
.25	21	17.9	26	21.8	14	12.3	15	13.1
.5	19	16.4	31	25.7	14	12.1	18	15.1
1.0	19	16.3	58	47.0	17	14.6	34	27.9
1.5	38	31.3	171	136.1	19	16.3	59	47.6
2.0	135	107.7	nc		21	17.8	75	60.1
4.0	nc		nc		26	21.7	52	42.1
8.0	nc		nc		76	60.9	105	83.6

TABLE 1

Iterations and time (entire CPU time including time for algebraic reduction) on model problem with $\sigma = 0$ using 1 lexicographic Gauss-Seidel iteration per level. 'nc' indicates no convergence.

We first consider the model PDE previously int roduced

$$-\mathrm{Au} + \sigma u_x + \tau u_y = 0 \tag{12}$$

with zero Dirichlet boundary conditions on the unit square so that the exact solution is u = 0. σ and τ are varied to test the cyclic reduction approach over a wide range of parameters. Additionally, we experiment with different orderings within the Gauss-Seidel smoothing. Tables 1 and 2 depict the results using a Gauss-Seidel relaxation procedure and natural ordering. Specifically, the columns labeled 'DS' correspond to the case when the points are relaxed in the direction of the flow. That is, up stream grid points are relaxed first. The columns labeled 'AS' denote the case when the points are relaxed in the up stream direction. That is, opposite to the flow direction. For all the experiments in this section a random initial guess is used and the computation is terminated when the initial residual is reduced by 10^{10}. Additionally, all experiments were performed on a 64 x 64 grid. For the purpose of comparison we illustrate the multigrid behavior on both the reduced equations and the original equations as a function of the cell Reynolds numbers, 6 and y.

By comparing the two schemes we can see the computational benefits of cyclic reduction. Specifically, when δ and y are less than 1.0, the cyclic reduction scheme

Y	original				reduced			
	DS		AS		DS		AS	
	iter	time	iter	time	iter	time	iter	time
0.	23	19.5	23	19.5	14	12.3	14	12.3
.25	21	17.9	30	24.9	12	10.7	17	14.7
.5	16	12.7	47	38.3	11	10.0	21	17.8
1.0	1	2.0	194	153.9	1	2.0	40	32.7
1.5	nc		nc		13	11.6	51	41.3
2.0	nc		nc		19	16.2	48	39.0
4.0	nc		nc		68	54.6	81	64.8
8.0	nc		nc		225	177.7	249	196.6

TABLE

Iterations and time (entire CPU time including time for algebraic reduction) on model problem with $\sigma = \tau$ using 1 lexicographic Gauss-Seidel iteration per level.

converges noticeably faster than the standard scheme and thus requires less overall time.' Further, when δ and y are greater than 1.0, the multigrid method applied to the original equations usually does not converge. That is, the Gauss-Seidel procedure does not smooth the error. By contrast, the multigrid convergence for the reduced equations is quite satisfactory for all values of 6. For both schemes the convergence behavior is quite sensitive to the ordering of the equations. Specifically, when the equations are ordered in the direction of the flow, the convergence rate is dramatically improved as compared to when the points are ordered opposite the flow direction.

In many applications the flow direction is not known or the flow is quite complicated. Further, on parallel machines it is somewhat difficult to efficiently parallelize the somewhat sequential Gauss-Seidel scheme with natural ordering. In these cases, it is usually best to use a multicolor ordering strategy. In particular, for a 5 point discretization where the grid points are colored in a red-black (checkerboard) fashion, it is possible to upc all the red points in parallel and then to update all the black

[4] The cyclic reduction code was developed from a standard multigrid code and has not been optimized. Currently, some useless operations are performed. We anticipate that the cyclic reduction run time can be reduced by 25% with further optimizations.

γ	original		reduced	
	iter	time	iter	time
0.	16	18.6	9	9.5
.25	15	17.5	11	11.2
.5	15	16.3	13	12.7
1.0	34	37.5	22	21.4
1.5	130	132.1	33	32.9
2.0	nc		41	39.0
4.0	nc		31	32.1
8.0	nc		94	89.2

TABLE 3

Iterations and time (entire CPU time including time for algebraic reduction) on model problem with $\sigma = 0$ using 1 multicolor Gauss-Seidel iteration on each level.

points in parallel. Since the points with the same color are independent of each other, the convergence is not particularly sensitive to the flow direction. In the case of a 9 point discretization one can use 4 colors and a 4 color relaxation scheme to achieve the same affect. In Tables 3 and 4 we illustrate the performance using the multicolor ordering schemes on the model problem. Once again it is quite clear that the reduced operator performs significantly better than the original operator and that unlike the original operator convergence occurs for all 6 and y. Overall, the multicolor schemes usually do not perform as well as the downstream orderings for highly convective flows. However, they perform much better than the upstream orderings and thus are probably the best choice when the flow direction is not known or when the computations are carried out on a parallel computer.

We now consider a series of nonconstant coefficient problems. First we consider the following convection diffusion PDE taken from Van der Vorst, 1981:

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

with Dirichlet boundary conditions. The CPU time and iterations are illustrated in Table 5. The most important observation is that the results are similar to the constant coefficient case. Specifically, the solution time for the cyclically reduced case is significantly better than when the multigrid method is applied to the original

γ	original		reduced	
	iter	time	iter	time
0.	16	18.6	9	9.5
.25	15	15.2	13	13.3
.5	20	19.9	18	17.7
1.0	165	154.0	33	33.0
1.5	nc		38	36.9
2.0	nc		33	30.9
4.0	nc		74	67.5
8.0	nc		270	243.1

TABLE 4

Iterations and time (entire CPU time including time for algebraic reduction) on model problem with $\sigma = \tau$ using 1 multicolor Gauss-Seidel iteration on each level.

smoother	original		reduced	
	iter	time	iter	time
1 multicolor	24	22.0	10	8.9
2 multicolor	13	15.5	8	9.1
1 lexicographical	31	24.5	14	11.1
2 lexicographical	15	14.2	10	9.4

TABLE 5

Number of multigrid iterations using different Gauss-Seidel smoothing combinations on each grid for the Van der Vorst problem.

equations. It is interesting to note that when only one relaxation sweep is performed per level that the convergence benefit of the cyclic reduction procedure is greater than when 2 **Gauss-Seidel** sweeps are used. This is due to the fact that when 2 sweeps **are** used the bottleneck in the convergence rate is no longer the smoothing rates but instead the accuracy of the coarse grid correction. Thus, the cyclic reduction benefit is somewhat less. I. terms of overall work, the best approach is the cyclic reduction scheme that uses only one Gauss-Seidel iteration.

γ	original		reduced	
	iter	time	iter	time
0.	13	32.8	6	16.1
20.	13	32.4	6	15.7
40.	12	30.7	7	17.5
60.	11	28.2	8	20.5
80.	11	28.6	8	20.5
100.	17	43.9	8	20.6
120.	nc		9	23.1
140.	nc		9	23.0

TABLE 6

Iterations and time (entire CPU time including time for algebmic reduction) on (12) using 6 multicolor Gauss-Seidel iteration on each level. 'nc' indicates no convergence.

We next consider the problem

$$-[b(x,y)u_x]_x - [c(x,y)u_y]_y + [b(x,y)\,c(x,y)u]_x$$

(13)

$$+ [d(x,y)u]_y + e(x,y)u = f$$

defined on the unit square with

$$b(x, y) = e^{-xy}, \qquad c(x, y) = x + y,$$
$$d(x,y) = \gamma(x + y), \qquad e(2, y) = (1 + xy)^{-1}$$

with zero Dirichlet boundary conditions. In general, as y increases the problem becomes more convective. In Table 6 we illustrate the results as a function of y. Once again the cyclic **reduction/multigrid** often produces an answer in less than half the time required for the traditional scheme. Further, for large enough y, we see that standard multigrid does not converge while the cyclic **reduction/multigrid** procedure converges satisfactorily.

We conclude with a circulating flow problem taken from [12]

$$-\epsilon\Delta u + 2y(1 - x^2)u_x - 2x(1 - y^2)u_y \text{ defined on } \Omega = (-1, 1) \times (0, 1)$$

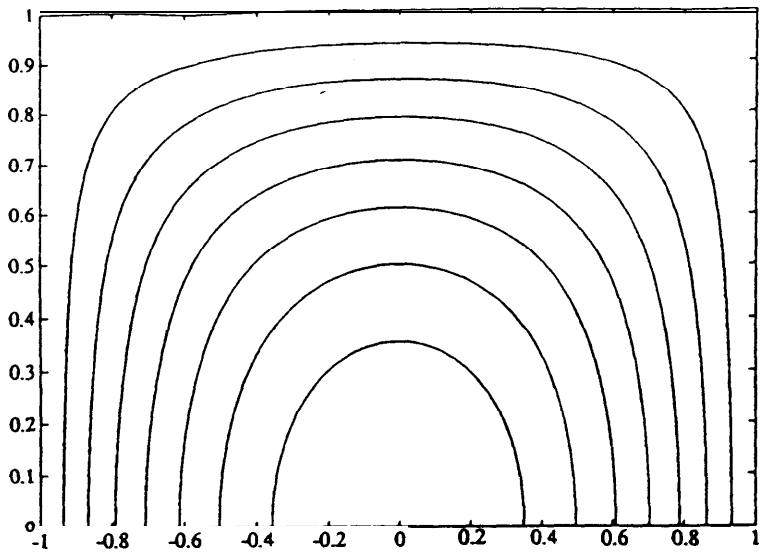

FIG. 7. *Stream fines* for circulation *flow problem*.

with

$$
\begin{aligned}
u &= 0 && 0 \le y \le 1, x = \pm 1, \\
u &= 0 && -1 \le 2 \le 1, y = 1, \\
u_n &= 0 && 0 \le x \le 1, y = 0, \\
u &= (1 - x^2) && -1 \le x < 0, y = 0,
\end{aligned}
$$

These equations essentially describe a circular flow entering (leaving) the cavity at $y = 0$ and $x < 0.0$ ($x > 0.0$). Contour lines are given in Figure 7 corresponding to the case of $\epsilon = .001$ and $h = \frac{1}{32}$ where h is the grid spacing in the x and y directions. In Table 7, the results are given for a multigrid iteration using 4 multicolor Gauss-Seidel iterations for each grid level. From the table, we see that the convergence for the reduced equations is often significantly faster than for the original. Further, the multigrid method applied to the cyclically reduced equations always converges whereas the method diverges when cyclic reduction is not used on the highly convective flows.

7. Conclusions. A new method has been given for convection-diffusion equations. The procedure, cyclic reduction/multigrid, proceeds by algebraically eliminating half the unknowns and then solves for the remaining unknowns using a multigrid method. The primary benefits of the algebraic reformulation is. the addition of dissipative terms in the corresponding PDE, the coarsening of the grid, and the smooth

ϵ	original		reduced	
	iter	time	iter	time
10.	12	19.2	8	7.9
1.	12	18.9	8	7.9
.1	11	11.4	8	7.9
.01	nc	-	13	12.0
.005	nc	-	15	13.6
.001	nc	-	159	134.5

TABLE

Iterations and time (entire CPU time including time for algebraic reduction) on circulating flow problem using 2 multicolor Gauss-Seidel iteration on each level. 'nc' indicates no convergence.

spreading of the discrete operator (9 point as opposed to 5 point). For dissipative equations, the smooth spreading of the stencil is the most important. For highly convective equations, the increased dissipation is most important. A Fourier analysis has been used to illustrate the improvement in the smoothing rates associated with the reduced operator. Additionally, a number of numerical experiments have been given to demonstrate the computational efficiency of the procedure. In general, when the PDE is not strongly convective, there is a noticeable improvement in the convergence rate associated with the reduced operator. Further, when the equations are highly convective or unstable, the cyclic **reduction/multigrid** method often converges quite satisfactorily while the standard multigrid method may diverge.

REFERENCES

[1] O. AXELSSON AND I. GUSTAFSSON, *On the use of preconditioned conjugate gmdient methods for red-black order* five-point *difference schemes,* J Comp Phys, 35 (1980), pp. 284–289.

[2] A. BRANDT, *Multi-level adaptive solutions to boundary-value problems,* Math. Comp., 31 (1977), pp. 333–390.

[3] ——, *Multigrid techniques: 1984 guide, with applications to fluid dynamics,* Tech. Rep. Nr. 85, GMD-AIW, Postfach 1240, D5205, GMD Studien, St. Augustin 1, W. Germany, 1984.

[4] W. BRIGGS, *Multigrid Tutorial,* SIAM, Philadelphia, 1987.

[5] H. ELMAN AND G. GOLUB, *Iterative methods for cyclically reduced non-self-adjoint* linear *systerns,* Tech. Rep. NA-89-01, Stanford University, Stanford, Ca, Feb. 1989.

[6] ——, *Iterative methods for cyclically reduced non-self-adjoint linear systems* ii, Tech. Rep. NA-89-07, Stanford University, Stanford, Ca, June 1989.

[7] ——, *Line iterative methods for cyclically reduced discnte convection-diffusion problems,* Tech. Rep. NA-90-01, Stanford University, Stanford, Ca, Feb. 1990.

[8] R. GLOWINSKI, Y. KUZNETSOV, G. MEURANT, J. PERIAUX, AND O. WIDLUND, eds., Domain Decomposition Methods *for* Partial *Differential Equations IV,* SIAM, Philadelphia, 1991.

[9] W. HACKBUSCH, *Multi-grid Methods and Applications,* Springer-Verlag, Berlin, 1985.

[10] D. JESPERSEN, *Multigrid methods for partial differential equations,* in Studies in Numerical Analysis, G. Golub, ed., vol. *24* of MAA Studies in Mathematics, 1984, pp. 270–318.

[11] R. KETTLER, *Analysis and* wmparison *of relaxation schemes* in *robust multigrid and preconditioned conjugate gmdient methods,* in Multigrid Methods, W. Hackbusch and U. Trottenberg, eds., vol. 960 of Lecture Notes in Math., Springer-Verlag, 1982, pp. 502–534.

[12] K. W. MORTON, *Generalised Galerkin methods* for *steady and unsteady problems,* in Numerical Methods for Fluid Dynamics, K. W. Morton and M. J. Baines, eds., Orlando, 1982, Academic Press.

[13] R. B. SIMPSON, *Testing for effects of asymmetry and instability on preconditioned iterations of* conjugate gmdient type, Tech. Rep. CS-91-22, University of Waterloo, Waterloo, Ontario, May 1991.

[14] D. YOUNG, *Iterative* Solution *of Large Linear* Systems, Academic Press, New York, 1971.

Appendix 1: Continuous Derivation of Reduced Operator

We derive the interior stencil for the reduced system of equations corresponding to applying one step of cyclic reduction to a standard S-point discretization of

$$-\mathrm{Au} + \sigma u_x + \tau u_y = f. \tag{14}$$

Specifically, we consider the following differential equation:

$$\begin{aligned}
&(1 + \frac{h^2\sigma^2}{8})u_{xx} + (1 + \frac{h^2\tau^2}{8})u_{yy} + \mathrm{au}, + \tau u_y + \frac{\sigma\tau h^2}{4}u_{xy} + \\
&\frac{h^2}{4}(\sigma u_{yyx} + \tau u_{xxy} + \sigma u_{xxx} + \tau u_{yyy}) + \\
&\frac{h^2}{8}(u_{xxxx} + 2u_{xxyy} + u_{yyyy}) \;=\; f + \frac{h^2}{8}(\sigma f_x + \tau f_y + f_{xx} + f_{yy})
\end{aligned} \tag{15}$$

obtained by differentiating (14) multiple times and then taking linear combinations. We can now consider approximating (15) using a finite difference operator. To do so, we first define a series of stencils and give their leading error terms.

$$\begin{aligned}
D_{xx}^{(2h)}u &= \frac{1}{4h^2}[u(x+2h,y) - 2u(x,y) + u(x-2h,y)] \\
&\approx u_{xx} + \frac{h^2}{3}u_{xxxx} \\
D_{yy}^{(2h)}u &= \frac{1}{4h^2}[u(x,y+2h) - 2u(x,y) + u(x,\mathrm{Y}-2h)] \\
&\approx u_{yy} + \frac{h^2}{3}u_{yyyy} \\
D_{x}^{(2h)}u &= \frac{1}{4h}[u(x+2h,\mathrm{y}) - u(x-2h,y)] \\
&\approx u_x + \frac{2h^2}{3}u_{xxx} \\
D_{\mathrm{Y}}^{(2h)}u &= \frac{1}{4h}[u(x,\mathrm{y}+2h) - u(x,\mathrm{y}-2h)] \\
&\approx u_y + \frac{2h^2}{3}u_{yyy} \\
\hat{D}_{x}^{(h)}u &= \frac{1}{4h}[u(x+h,y+h) + u(x+h,y-h) \\
&\quad -u(x-h,y+h) - u(x-h,y-h)] \\
&\approx u_x + \frac{h^2}{6}u_{xxx} + \frac{h^2}{2}u_{yyx}
\end{aligned}$$

$$
\begin{aligned}
\hat{D}_y^{(h)}u &= \frac{1}{4h}[u(x+h, y+h) - u(x+h, y-h) \\
&\quad + u(x-h, y+h) - u(x-h, y-h)] \\
&\approx u_y + \frac{h^2}{6}u_{yyy} + \frac{h^2}{2}u_{xxy} \\
(\hat{D}_{xx}^{(h)} + \hat{D}_{yy}^{(h)})u &= \frac{1}{2h^2}[u(x+h, y+h) + u(x+h, y-h) \\
&\quad - 4u(x,y) + u(x-h, y+h) + u(x-h, y-h)] \\
&\approx u_{xx} + u_{yy} + \frac{h^2}{12}[u_{xxxx} + u_{yyyy}] + \frac{h^2}{2}u_{xxyy} \\
D_{xy}^{(h)}u &= \frac{1}{4h^2}[u(x+h, y+h) - u(x+h, y-h) \\
&\quad - u(x-h, y+h) + u(x-h, y-h)] \\
&\approx u_{xy} + \frac{h^2}{6}[u_{xxxy} + u_{yyyx}].
\end{aligned}
$$

We approximate the first and second order terms in (15) using the difference operators defined above. This yields the mixed **difference/differential** equation

$$
\begin{aligned}
&(\frac{1}{2} + \frac{h^2\sigma^2}{8})D_{xx}^{2h}u + (\frac{1}{2} + \frac{h^2\tau^2}{8})D_{yy}^{2h}u + \frac{1}{2}(\hat{D}_{xx}^{(h)} + \hat{D}_{yy}^{(h)})u + \\
&\frac{\sigma}{2}(D_x^{(2h)}u + \hat{D}_x^{(h)}u) + \frac{\tau}{2}(D_y^{(2h)}u + \hat{D}_y^{(h)}u) + \\
&\frac{\sigma\tau h^2}{4}D_{xy}^{(h)} + h^2[\frac{1}{12}u_{xxxx} + \frac{1}{12}u_{yyyy} + \frac{\sigma}{6}u_{xxx} + \frac{\tau}{6}u_{yyy}] \\
&\qquad \approx f + \frac{h^2}{8}(\sigma D_x^{(h)}f + \tau D_y^{(h)}f + D_{xx}^{(h)}f + D_{yy}^{(h)}f)
\end{aligned}
$$

where we have combined the $O(h^2)$ error terms in the difference approximations with the $O(h^2)$ terms in (15). The main point is that the discrete **operator** defined by the difference **operators** on the right hand side of the above equation is identical to the discrete operator corresponding to performing one step of cyclic reduction to a standard S-point **discretization** of (14). Further, the $0(h^2)$ error tern the above **expression** are identical to the $O(h^2)$ error terms that would be **obtained** one approximates the original convection-diffusion equation by central differences.

Appendix 2: Multicolor Gauss-Seidel Smoothing Analysis

In this section we sketch the basic elements of the multicolor Gauss-Seidel smoothing analysis. To begin, we express the Gauss-Seidel operator as a series of Jacobi, projection, and interpolation operations. Specifically, consider the projection operators

$$\begin{aligned} w^{ee} &= R_{ee}v, \\ w^{eo} &= R_{eo}v, \\ w^{oe} &= R_{oe}v, \\ w^{00} &= R_{oo}v \end{aligned} \tag{16}$$

which each map v (defined on the fine grid) to a coarse grid. Specifically, the w's are defined on coarser grids and assume the same values as v. For example, w^{ee} is defined on the even points in both the x and y directions of the fine grid and takes on the same values as v on these even points; w^{eo} is defined on the even points in the x direction and the odd points in the y direction and assumes the same value of v at these points; etc. We also define the Jacobi operator, J, as

$$J = I - D^{-1}A$$

where A is the discretization operator and D is the diagonal of A. We can now write the four color Gauss-Seidel iteration matrix as

$$G = (I - X_{ee} + X_{ee}J)(I - X_{oo} + X_{oo}J)(I - X_{eo} + X_{eo}J)(I - X_{oe} + X_{oe}J)$$

where

$$X_{ij} = R_{ij}^T R_{ij}.$$

That is, we can view the first step of a 4 color Gauss-Seidel operator as a Jacobi iteration that is only performed on the coarse grid points defined by X_{ij}. To analyze the smoothing properties of the multicolor Gauss-Seidel operator we compute the largest eigenvalue of the operator

$$S = CG$$

where C is an approximation to the multigrid coarse grid correction. Specifically, C is defined as

$$Cv_{kj} = \begin{cases} 0 & \text{if } |k| \le n/4, \quad |j| \le n/4 \\ v_{kj} & \text{otherwise} \end{cases}$$

where v_{kj} is the vector

$$\{v_{kj}\}_{lm} = e^{2\pi i(kx_l + jy_m)} \qquad -\frac{n}{2} < k, j \leq \frac{n}{2}$$

corresponding to a Fourier mode,

$$x_l = \frac{l}{n}, \qquad y_m = \frac{m}{n} \qquad l, m = 0, \ldots . n-1,$$

and n is the number of mesh points in each dimension of the grid. This definition simple states that C completely eliminates low frequency Fourier modes and does not disturb high frequency modes.

To compute the eigenvalues of S, we first take the Fourier transform

$$\tilde{S} = \tilde{C}(I - \tilde{X}_{ee} + \tilde{X}_{ee}\tilde{J})(I - \tilde{X}_{oo} + \tilde{X}_{oo}\tilde{J})(I - \tilde{X}_{eo} + \tilde{X}_{eo}\tilde{J})(I - \tilde{X}_{oe} + \tilde{X}_{oe}\tilde{J})$$

where we use a tilde to denote the Fourier transform of each operator. For constant coefficient discretizations and periodic boundary conditions, J is a circulant matrix and hence is diagonalizable by the Fourier transform. By definition $\tilde{C}$ is also diagonal. Similarly, the X_{ij} operators reduce to block diagonal matrices (with 4 × 4 blocks) by the Fourier transform. This can be seen by exmaining the Fourier properties of the projection operators. Each R_{ij} projects four Fourier modes onto one coarse grid Fourier mode. For example,

$$\begin{aligned}
R_{ee}v_{kj} &= \hat{v}_{kj} \quad \text{if } |k| \leq n/4, \quad |j| \leq n/4, \\
R_{ee}v_{\hat{k}j} &= \hat{v}_{kj} \quad \text{if } |k| > n/4, \quad |j| \leq n/4, \\
R_{ee}v_{k\hat{j}} &= \hat{v}_{kj} \quad \text{if } |k| \leq n/4, \quad |j| > n/4, \\
R_{ee}v_{\hat{k}\hat{j}} &= \hat{v}_{kj} \quad \text{if } |k| > n/4, \quad |j| > n/4
\end{aligned}$$

where

$$\hat{k} = sgn(k)\frac{n}{2} - k, \qquad \hat{j} = sgn(j)\frac{n}{2} - j,$$

$$sgn(k) = \begin{cases} 1 & \text{if } k \geq 0 \\ -1 & \text{if } k < 0 \end{cases}$$

and $\hat{v}_{kj}$ are the Fourier modes on the (even-even) coarse grid.

We omit the details and simply state that the smoothing operator, $\tilde{S}$, can be transformed to block diagonal (4 x 4 blocks) using a simple permutation matrix. That

is, the 4 x 4 blocks can be compute analytically. The maximum eigenvalue of S is given by the maximum eigenvalue of all the block matrices. In specific cases, explicit expressions can be computed. However, for the general operators considered in this paper we have used a program to compute the maximum eigenvalue of these 4 x 4 blocks and obtain the smoothing rates given in Section 5