

ARGONNE NATIONAL LABORATORY  
9700 South Cass Avenue  
Argonne, Illinois 60439

## **THE EFFECT OF MANY-COLOR ORDERINGS ON THE CONVERGENCE OF ITERATIVE METHODS \***

**Mark T. Jones and Paul E. Plassmann**

Mathematics and Computer Science Division  
Preprint MCS-P292-0292  
May 1992

### **ABSTRACT**

It is well known that orderings based on graph colorings can slow the convergence of many iterative methods. However, orderings based on graph colorings can be used to build scalable, parallel iterative methods. For example, SOR with consistent orderings, SSOR, and incomplete Cholesky can all be implemented in a scalable manner using graph colorings. It has been observed experimentally that if, for a fixed problem, the number of colors used is increased, then the SSOR PCG method will converge faster. In this paper we prove that for a model problem the multilevel SOR algorithm with consistent orderings will converge more quickly as the number of colors used is increased. We also give a formula for computing the optimal relaxation parameter and rate of convergence as a function of the number of colors. We prove a similar, but limited, result for ICCG.

These results are useful because they allow the algorithm designer to trade off parallel efficiency for fewer iterations. For example, the nine-point Laplacian can be colored using a minimum of four colors. However, a matrix ordering based on four-colorings may result in poor convergence. If eight rather than four colors are used, the convergence rate will be increased without, perhaps, unduly affecting the parallel efficiency.

**Key words:** Fourier analysis, ICCG algorithm, multicoloring, parallel algorithms, SOR algorithm, sparse matrices

**AMS(MOS) subject classifications:** 65F10, 65N22, 65Y05

\* This work was supported by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy, under Contract W-31-109-Eng-38.

**1. Introduction.** Many iterative methods exhibit excellent parallel efficiency if the matrices have been reordered according to a graph coloring [13]. However, for most of these iterative methods, the convergence rate after reordering is slower than that of the “natural” ordering. For example, for the standard five-point difference operator the incomplete Cholesky preconditioner has a condition number of  $\frac{2}{(2+\sqrt{2})\pi^2 h^2}$  with the natural ordering [2] and a condition number of  $\frac{1}{\pi^2 h^2}$  with a red/black ordering [8].

Harrar [5] has observed experimentally for the SSOR PCG method that the convergence rate improves if more colors are used to color the graph of the matrix. Using the natural ordering and the colorings described in §2, we show in Figure 1 the reduction in the residual for the ICCG method. These results are for the standard nine-point difference operator on a  $128 \times 128$  grid. We observe that as the number of colors is increased, the number of iterations necessary to achieve the same accuracy decreases. For large problems, one can use more colors with some

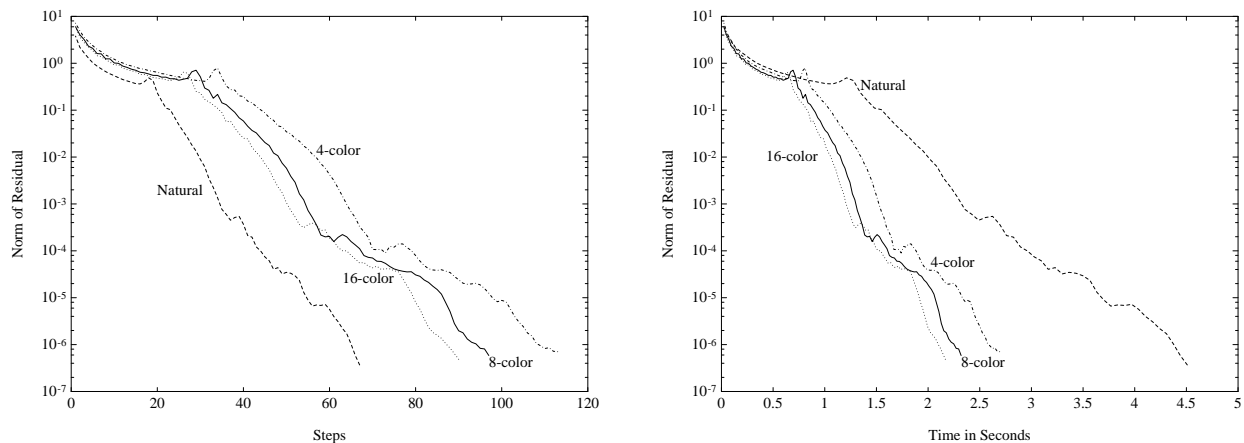


FIG. 1. *The convergence rate and the cost of ICCG with varying numbers of colors*

sacrifice of parallel efficiency and reduce solution time by reducing the number of iterations. For example, in the right plot in Figure 1 we show the predicted times associated with these multicolorings when we use a variant of the parallel execution model given by Elman and Agrón [3] for solving the same  $128 \times 128$  grid problem with 16 processors. For the model, we chose the parameters  $\tau = 10$ ,  $c = 10$ , and  $s = 1$ ; the same parameters suggested by Elman and Agrón in §6 of their paper.

For two different iterative methods, a stationary method and a preconditioned method, we use Fourier analysis techniques to prove that increasing the number of colors will result in faster convergence. This analytic information can be used by practitioners to make algorithmic design trade-offs on high-performance architectures.

The first method that we examine is a multilevel version of the red/black SOR method [15]: a generalization suggested by Kuo and Levy [9]. By using a multilevel

approach, a *consistent ordering* can be obtained for any number of colors. The optimal relaxation factor can be cheaply computed for any consistently ordered, symmetric positive definite matrix [4]. Consider the four-color example given in Figure 2. At the outermost level of the multilevel algorithm, the four colors are split into two groups to form a  $2 \times 2$  block matrix. This  $2 \times 2$  matrix is trivially consistently ordered, and an optimal relaxation parameter can be computed. At the second level, each block is split into two matrices to form  $2 \times 2$  block matrices for which optimal relaxation parameters can be computed. At the innermost level, the matrices are diagonal. Given that each processor will have approximately

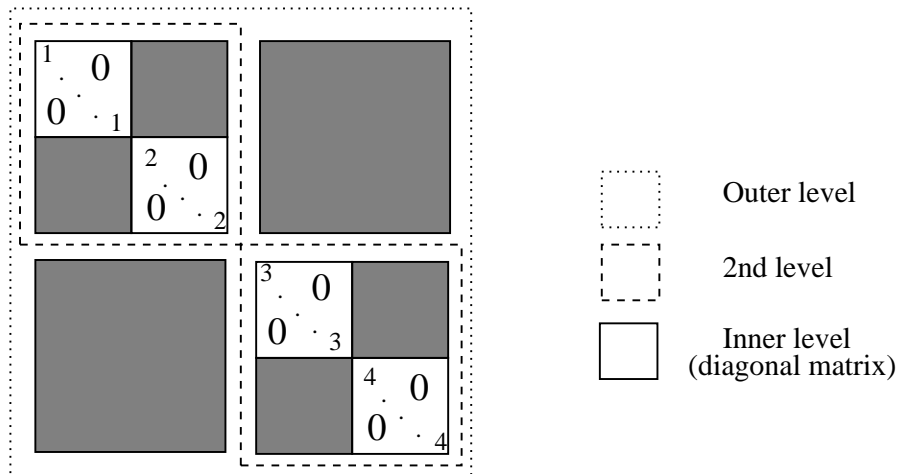


FIG. 2. A multilevel SOR algorithm for four colors

the same number of grid points of a given color, the load balancing throughout the algorithm will be quite good. Further discussion of this algorithm is given in §3. We note that for the specific case of four-coloring the nine-point Laplacian, an alternative to the multilevel SOR algorithm is a nonconsistently ordered SOR algorithm for which the optimal relaxation parameter can be determined by solving a quartic equation [1].

The multilevel SOR algorithm has advantages over the traditional approach of obtaining a consistent ordering by breadth-first search (BFS) and then using red/black SOR. In the BFS approach, subproblems associated with narrow strips of grid points must be solved. For the large matrices that must be solved on parallel computers, these subproblems cannot fit onto a single processor. For example, if we wish to solve a symmetric matrix generated from a finite difference stencil on a cube, the subproblems are planes of the cube. The subproblem size increases as the cube size is increased, but the memory on a single processor is fixed; for a scalable parallel implementation the subproblems must be split across processors. Therefore, one is faced once again with the problem of solving sparse matrices in parallel. In the multilevel algorithm, the parallel solution of the subproblems is inherent in the algorithm. The only operations required are sparse matrix by

vector multiplication and the solution of the diagonal submatrices; both operations are parallel.

The BFS algorithm has an additional drawback when solving unstructured problems in parallel. The parallel determination of the subproblem strips is difficult. In general, no good parallel method for BFS exists; therefore, the ordering of the nodes can take an inordinate amount of time on a parallel computer. In contrast, good parallel graph coloring heuristics exist. Recently Jones and Plassmann have given a scalable, parallel heuristic for finding colorings for unstructured, sparse matrices [7].

In addition to the multilevel SOR algorithm, we analyze the conjugate gradient algorithm preconditioned by an incomplete Cholesky factorization [11, 12]. Jones and Plassmann have given scalable, parallel methods for the ICCG algorithm for solving unstructured, sparse, symmetric positive definite matrices [6].

The remainder of the paper is organized as follows. In §2 we analyze the convergence properties of the multilevel SOR algorithm with varying numbers of colors. We further discuss the multilevel algorithm and give experimental results in §3. In §4 we give a similar but more limited analysis for ICCG.

**2. Analysis of Multilevel SOR.** In this section, we use Fourier analysis to find the spectral radius of the multilevel SOR algorithm for the five-point Laplacian operator for any number of colors. This general analysis is restricted to a particular coloring pattern. We also give a shorter, restricted proof for a different coloring pattern for which the outer iteration uses blocks rather than strips. We believe that the extension of these results to other operators and coloring patterns is straightforward in many cases. More important, we have found these results to hold true in practical, complicated problems.

Consider an  $N \times N$  grid, whose nodes are colored by using the following pattern.

$$(2.1) \quad \begin{array}{cccccc} n & 1 & 3 & 5 & & n-1 & 2 & 4 \\ n-1 & 2 & 4 & 6 & \cdots & n & 1 & 3 \\ n & 1 & 3 & 5 & \cdots & n-1 & 2 & 4 \\ n-1 & 2 & 4 & 6 & & n & 1 & 3 \end{array}$$

We index the  $x$ -direction by  $j$  and the  $y$ -direction by  $k$ . For convenience we use the notation

$$(2.2) \quad \begin{aligned} S_{j,\eta} &= \sin(j\eta\pi h) \\ C_{j,\eta} &= \cos(j\eta\pi h), \end{aligned}$$

where  $h = 1/(N+1)$ . Let  $x_{j,k}$  be the discretization of the scalar function  $x$  at the point  $(j, k)$  on the grid. If the color of the grid point  $(j, k)$  is  $c$ , we denote this point by  $x_{j,k}^{(c)}$  and define the Fourier transform of this point by the equation

$$(2.3) \quad x_{j,k}^{(c)} = \sum_{(\eta,\xi) \in K^{(c)}} \hat{x}_{\eta,\xi}^{(c)} S_{j,\eta} S_{k,\xi}.$$

We do not explicitly compute the set  $K^{(c)}$  here; we only note that the lowest frequency pair included is  $(\eta, \xi) = (1, 1)$ .

Consider the action of the standard five-point Laplacian operator  $\Delta$  on the scalar function  $x$  at grid point  $(j, k)$ :

$$(2.4) \quad -\frac{h^2}{4}\Delta(j, k)x = x_{j,k} - \frac{1}{4}(x_{j-1,k} + x_{j+1,k} + x_{j,k-1} + x_{j,k+1}).$$

In the frequency domain, we note that the transformed Laplacian operator is approximately block diagonal, with a block  $\hat{\Delta}(\eta, \xi)$  for each frequency pair  $(\eta, \xi)$ . By “approximately” we mean that for more than four colors, there is a term involving  $S_{1,\eta}C_{j,\eta}S_{k,\xi}$  that does not cancel in the Fourier representation of Equation 2.4. However, we are interested only in the low-frequency limit. For fixed  $\eta$ , we have that  $S_{1,\eta} \rightarrow 0$  as  $N \rightarrow \infty$ . In this limit, which we assume for the rest of this section, these blocks can be expressed as

$$(2.5) \quad \hat{\Delta}(\eta, \xi) = \begin{bmatrix} \hat{M}_n(\eta, \xi) & \hat{N}_n(\eta, \xi) \\ \hat{N}_n^T(\eta, \xi) & \hat{M}_n(\eta, \xi) \end{bmatrix}.$$

This block operator acts on the vector  $\hat{x}(\eta, \xi)$  given by

$$(2.6) \quad \hat{x}(\eta, \xi) = \begin{bmatrix} \hat{x}_{\eta,\xi}^{(1)} \\ \hat{x}_{\eta,\xi}^{(2)} \\ \vdots \\ \hat{x}_{\eta,\xi}^{(n)} \end{bmatrix}.$$

Substituting Equation 2.3 into Equation 2.4, we are able to compute the two  $n/2 \times n/2$  matrices  $\hat{M}_n(\eta, \xi)$  and  $\hat{N}_n(\eta, \xi)$

$$(2.7) \quad \hat{M}_n(\eta, \xi) = \begin{bmatrix} 1 & \alpha_\eta & \beta_\xi & & & \\ \alpha_\eta & 1 & 0 & \beta_\xi & & \\ \beta_\xi & 0 & 1 & \alpha_\eta & \beta_\xi & \\ 0 & \beta_\xi & \alpha_\eta & 1 & 0 & \ddots \\ & & \beta_\xi & 0 & \ddots & \beta_\xi \\ & & & \ddots & & 1 & \alpha_\eta \\ & & & & \beta_\xi & \alpha_\eta & 1 \end{bmatrix},$$

$$(2.8) \quad \hat{N}_n(\eta, \xi) = \begin{bmatrix} & & & & 0 & \beta_\xi \\ & & & & \beta_\xi & 0 \\ & & & 0 & & \\ & & & 0 & & \\ & & 0 & & & \\ & \beta_\xi & 0 & & & \\ 0 & 0 & \beta_\xi & & & \end{bmatrix},$$

where  $\alpha_\eta = -C_{1,\eta}/2$  and  $\beta_\xi = -C_{1,\xi}/4$ .

We consider only the outer level of the multilevel SOR iteration by grouping all grid points colored by the first  $n/2$  colors into a first group, and the remainder of the grid points into a second group. For example, with  $n = 8$ , we have the following partition of the grid.

$$(2.9) \quad \begin{array}{c} 8 \\ 7 \\ 8 \\ 7 \end{array} \left| \begin{array}{cc} 1 & 3 \\ 2 & 4 \\ 1 & 3 \\ 2 & 4 \end{array} \right| \begin{array}{c} 5 \\ 6 \\ 5 \\ 6 \end{array} \left| \begin{array}{cc} 7 & 2 \\ 8 & 1 \\ 7 & 2 \\ 8 & 1 \end{array} \right| \begin{array}{c} 4 \\ 3 \\ 4 \\ 3 \end{array} \left| \begin{array}{c} 6 \\ 5 \\ 6 \\ 5 \end{array} \right.$$

Note that the grid is partitioned into strips; for  $n = 4$  we obtain the equivalent of the standard line red/black ordering.

This multilevel SOR iteration is consistently ordered. Thus, we can compute [15] the optimal relaxation parameter,  $\omega^*$ , for SOR from the spectral radius,  $\rho_J$ , of the Jacobi iteration matrix from the equation

$$(2.10) \quad \omega^* = \frac{2}{1 + \sqrt{1 - \rho_J^2}}.$$

The computation of  $\rho_J$  is more straightforward in the frequency domain. Because of the special structure of  $\hat{M}_n$  and  $\hat{N}_n$  and the approximate block diagonal structure of the Jacobi iteration matrix, we find

$$(2.11) \quad \rho_J \approx \max_{(\eta, \xi)} \rho(\hat{M}_n^{-1}(\eta, \xi) \hat{N}_n(\eta, \xi)).$$

It is clear that the maximum in Equation 2.11 is obtained for the lowest frequency mode,  $(\eta, \xi) = (1, 1)$ , where the block diagonal approximation is most accurate.

To simplify the notation in the following discussion, let  $M_n = M_n(1, 1)$ ,  $N_n = N_n(1, 1)$ ,  $\beta = \beta_1$ , and  $\alpha = \alpha_1$ . For  $n = 2^{i+1}$ , we define  $\rho_i = \rho(\hat{M}_n^{-1} \hat{N}_n)$ . It will be useful to define the following recursion relation:

$$(2.12) \quad \begin{aligned} s_1 &= \frac{\cos(\pi h)/2}{2 - \cos(\pi h)} \\ r_1 &= s_1 \\ s_{i+1} &= \frac{s_i^2}{1 - r_i^2} \\ r_{i+1} &= r_i(1 + s_{i+1}). \end{aligned}$$

**THEOREM 2.1.** *Let  $r_i$  and  $s_i$  be defined by the recursion relations shown in Equation 2.12. The spectral radius of the Jacobi iteration satisfies  $\rho_i = r_i + s_i$ .*

**Proof:** Suppose  $v$  is the eigenvector corresponding to the largest eigenvalue of  $\hat{M}_n^{-1} \hat{N}_n$ . By inspection, we note that the product  $\hat{N}_n v$  has the special form  $[x \ x \ 0 \ \cdots \ 0 \ x \ x]^T$ . Thus, the computation of the spectral radius is equivalent to computing the sum of four components of  $\hat{M}_n^{-1}$  multiplied by  $\beta$ ,

$$(2.13) \quad \rho_i = \left| \beta \left[ (\hat{M}_n^{-1})_{1,1} + (\hat{M}_n^{-1})_{1,2} + (\hat{M}_n^{-1})_{1,\frac{n}{2}-1} + (\hat{M}_n^{-1})_{1,\frac{n}{2}} \right] \right|.$$

The value of these components can be computed by an induction argument and use of the Sherman-Morrison-Woodbury formula

$$(2.14) \quad (A + UV^T)^{-1} = A^{-1} - A^{-1}U(I + V^T A^{-1}U)^{-1}V^T A^{-1}.$$

Let

$$(2.15) \quad \hat{M}_n = A + UV^T,$$

where the matrix  $A$  is

$$(2.16) \quad A = \begin{bmatrix} \hat{M}_{n/2} & 0 \\ 0 & \hat{M}_{n/2} \end{bmatrix},$$

and the matrices  $U$  and  $V$  are given by

$$(2.17) \quad U = \left[ \begin{array}{cc|cc} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array} \right], \quad V = \beta \left[ \begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right],$$

where  $U$  and  $V$  are  $\frac{n}{2} \times 4$  matrices. Here we show the middle 4 rows; the rest of these matrices are zero. Let  $S$  be the last two columns of  $\hat{M}_{n/2}^{-1}$ , and let  $\bar{S}$  be the first two columns of  $\hat{M}_{n/2}^{-1}$ . We note that the matrix  $\hat{M}_{n/2}^{-1}$  is persymmetric (i.e., symmetric about the “other” diagonal), so that  $\bar{S}$  is the persymmetric version of  $S$ . Also, let  $K$  be the leading principal  $2 \times 2$  matrix of  $\beta \hat{M}_{n/2}^{-1}$ . We compute the following matrices,

$$(2.18) \quad \begin{aligned} A^{-1}U &= \begin{bmatrix} 0 & S \\ \bar{S} & 0 \end{bmatrix} \\ A^{-1}V &= \beta \begin{bmatrix} S & 0 \\ 0 & \bar{S} \end{bmatrix} \\ V^T A^{-1}U &= \begin{bmatrix} 0 & K \\ K & 0 \end{bmatrix}. \end{aligned}$$

Continuing with our computation, we have

$$(2.19) \quad \left[ I + \begin{pmatrix} 0 & K \\ K & 0 \end{pmatrix} \right]^{-1} = \begin{bmatrix} K_e & -KK_e \\ -KK_e & K_e \end{bmatrix},$$

where

$$(2.20) \quad K_e = (I - K^2)^{-1}.$$

Combining these facts with Equation 2.14, we have

$$(2.21) \quad (A + UV^T)^{-1} = A^{-1} + \beta \begin{bmatrix} SKK_e S^T & -SK_e \bar{S}^T \\ -\bar{S}K_e S^T & \bar{S}KK_e \bar{S}^T \end{bmatrix}.$$

We observe that  $A$  is structured so that if we look at  $A$  as a block matrix with  $2 \times 2$  blocks, each of these  $2 \times 2$  blocks is symmetric and has equal diagonal elements. Therefore, each block, and in particular  $K$  and  $K_e$ , can be diagonalized when conjugated by the orthogonal matrix

$$(2.22) \quad Q = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}.$$

Thus, to compute  $\rho_i$  as given in Equation 2.13, we need only compute the largest diagonal elements of the upper left  $2 \times 2$  block and upper right  $2 \times 2$  block of  $(A + UV^T)^{-1}$  when conjugated by  $Q$ . We denote these values by  $r_i$  and  $s_i$ , respectively. Direct calculation establishes the recursion relation given in Equation 2.12.  $\square$

TABLE 1  
*A comparison of the predicted and calculated values of  $\rho_J$  for 8 colors*

Mesh Size	Prediction	Calculated
$4 \times 4$	0.51433	0.50965
$8 \times 8$	0.79573	0.79456
$16 \times 16$	0.93520	0.93511
$24 \times 24$	0.96919	0.96917
$32 \times 32$	0.98213	0.98213

In Table 1, by direct calculation of  $\rho_J$  we show that our low frequency Fourier representation of the Laplacian is quite accurate. The values for  $\rho_J$  predicted by Theorem 2.1 converge rapidly to the computed values. We note that Parter [14] obtains an approximation to the dominant eigenvalue  $\rho_R$  of the  $k$ -line Richardson method (equivalent to the Jacobi iteration) for elliptic partial differential equations with constant coefficients. For the Laplace equation with Dirichlet boundary conditions, this approximation is given by

$$(2.23) \quad \rho_R \approx 1 - \frac{k}{2} \Lambda h^2,$$

where  $\Lambda$  is the minimum eigenvalue of the Laplacian on the given domain. In our case, the domain is the unit square; thus we have  $\Lambda = 2\pi^2$ .

In Figure 3 we compare the estimate obtained by Parter in Equation 2.23 with the Fourier analysis estimate given by Theorem 2.1. The exact values for the spectral radius are indistinguishable from the Fourier values in this figure.

We note that a multilevel algorithm based on these colorings has the disadvantage that the inner iterations will require information exchange along the entire strip. For large problems it is unrealistic to expect that these strips will fit onto one processor. One can avoid interprocessor communication at all levels of the multilevel iteration, except the outer level, by using a “block” coloring. One possible



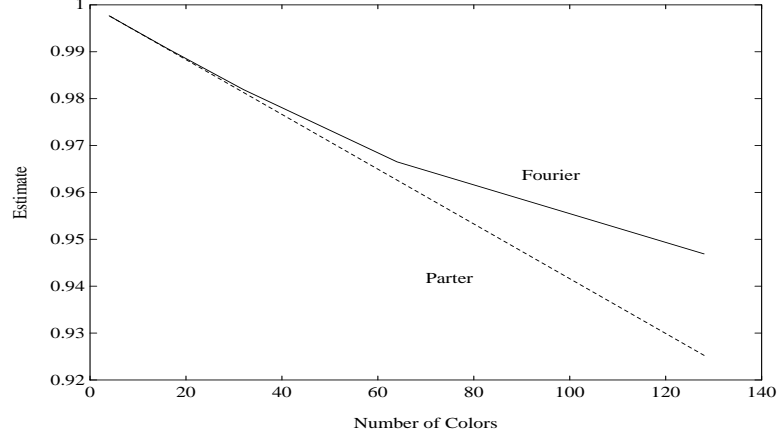


FIG. 3. Comparison of Parter's estimate for  $\rho_J$  with the Fourier estimate for a  $64 \times 64$  grid

block eight-coloring is the following.

$$(2.24) \quad \begin{array}{cccc} 1 & 4 & 5 & 8 \\ 3 & 2 & 7 & 6 \\ 5 & 8 & 1 & 4 \\ 7 & 6 & 3 & 2 \end{array}$$

Let  $\sigma_i$  be the spectral radius of the block Jacobi iteration corresponding to the outer level of block coloring with  $2^{i+2}$  colors. We obtain the following interesting result.

**THEOREM 2.2.** *We have that asymptotically  $\sigma_1 = \rho_1$ . That is, the spectral radius for the outer level of a block 8-coloring is asymptotically equivalent to line  $R/B$ .*

**Proof:** Consider the Fourier representation of the 5-point Laplacian for the coloring given in Equation 2.24:

$$(2.25) \quad \hat{\Delta}^B(\eta, \xi) = \begin{bmatrix} \hat{M}_n^B(\eta, \xi) & \hat{N}_n^B(\eta, \xi) \\ \hat{N}_n^{B^T}(\eta, \xi) & \hat{M}_n^B(\eta, \xi) \end{bmatrix}.$$

We compute that the matrices  $\hat{M}_n^B$  and  $\hat{N}_n^B$  have the form

$$(2.26) \quad \hat{M}_n^B(\eta, \xi) = \begin{bmatrix} 1 & 0 & \beta_\eta & \beta_\xi \\ 0 & 1 & \beta_\xi & \beta_\eta \\ \beta_\eta & \beta_\xi & 1 & 0 \\ \beta_\xi & \beta_\eta & 0 & 1 \end{bmatrix},$$

$$(2.27) \quad \hat{N}_n^B(\eta, \xi) = \begin{bmatrix} 0 & 0 & \beta_\eta & \beta_\xi \\ 0 & 0 & \beta_\xi & \beta_\eta \\ \beta_\eta & \beta_\xi & 0 & 0 \\ \beta_\xi & \beta_\eta & 0 & 0 \end{bmatrix},$$

where  $\beta_\eta = -C_{1,\eta}/4$  and  $\beta_\xi = -C_{1,\xi}/4$ .

Consider the computation of the spectral radius of the Jacobi iteration matrix for the lowest frequency mode. We note that conjugation by the block diagonal orthogonal matrix, with the diagonal blocks given by Equation 2.22, yields a system with spectral radius equivalent to that obtained from a line red/black analysis.  $\square$

TABLE 2  
*A comparison of the calculated values of  $\rho_J$  for line R/B and the block 8-coloring*

Mesh Size	Line R/B	Block 8-Coloring
$4 \times 4$	0.67928	0.66667
$8 \times 8$	0.88625	0.88515
$16 \times 16$	0.96652	0.96646
$24 \times 24$	0.98435	0.98434

This result is verified by the computational results shown in Table 2. Finally, we note that experimentally it can also be demonstrated that  $\sigma_i = \rho_i$  asymptotically for  $i > 1$ .

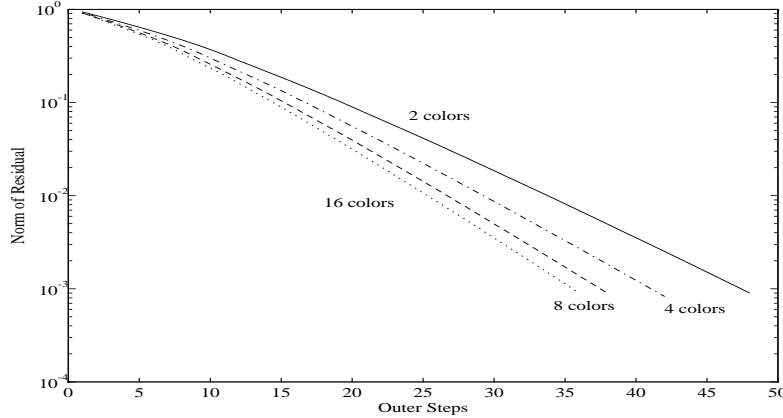


FIG. 4. *Convergence of different numbers of colors*

The improvement in convergence rates as the number of colors increases is not limited to the strip and block colorings. Experimentally, we have found the improvement with every coloring pattern that we have used. For example, if we use the following coloring pattern

$$(2.28) \quad \begin{array}{cccccccc} 1 & 2 & \cdots & \frac{n}{2} & \frac{n}{2} + 1 & \cdots & n-1 & n \\ \frac{n}{2} + 1 & \frac{n}{2} + 2 & \cdots & n & 1 & \cdots & \frac{n}{2} - 1 & \frac{n}{2} \\ 1 & 2 & \cdots & \frac{n}{2} & \frac{n}{2} + 1 & \cdots & n-1 & n \\ \frac{n}{2} + 1 & \frac{n}{2} + 2 & \cdots & n & 1 & \cdots & \frac{n}{2} - 1 & \frac{n}{2} \end{array}$$

on the five-point difference operator on a  $128 \times 128$  grid, we observe in Figure 4 that as the number of colors is increased, the convergence of the multilevel SOR algorithm improves.

**3. Results for Multilevel Algorithm.** In this section we briefly discuss the multilevel SOR algorithm when used with the block colorings described in the second part of §2. Following Kuo and Levy [9], we consider solving the inner subproblems iteratively; we may perform only a few iterations for each subproblem. We are now concerned with the convergence of the multilevel SOR algorithm when the inner subproblems are not solved to full accuracy.

From Lanzkron et al. [10] we know that the multilevel SOR algorithm can be characterized as an inner/outer iteration method. To guarantee convergence for a two-level inner/outer iteration method, the outer iteration must be a regular splitting and the inner iteration must be weak regular. For the model problem the splitting

$$\begin{aligned}
 A &= D - L - U = M - N \\
 M &= \omega^{-1}(D - \omega L)^{-1} \\
 N &= \omega^{-1}(\omega U + (1 - \omega)D)
 \end{aligned}
 \tag{3.29}$$

induced by the outer iteration is not a regular splitting because  $N$  is not positive when  $\omega > 1$ . However, we expect that if the inner problems are solved to reasonable accuracy, then the method will converge.

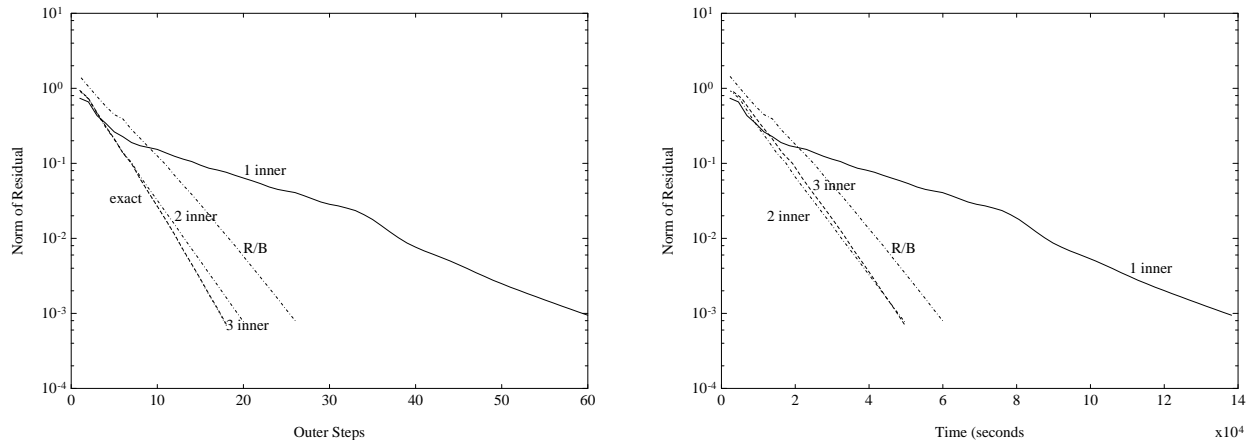


FIG. 5. Comparison of R/B SOR with 8-color multilevel SOR with varying numbers of inner iterations

In Figure 5 we show the convergence of the multilevel SOR algorithm with different numbers of inner iterations and compare this convergence with that of the standard red/black SOR algorithm for the 5-pt stencil. We also show that, given a parallel execution time model<sup>1</sup> similar to that used in Section 1, the multilevel SOR algorithm can have performance advantages over algorithms using fewer numbers

<sup>1</sup> The results are from a  $16 \times 16$  grid using 16 processors with model parameters  $s = 1$ ,  $\tau = 1000$ , and  $c = 0.1$ .

of colors. This advantage is magnified as the ratio of the message startup cost relative to the cost of computation increases.

**4. Analysis of Incomplete Factorization.** In this section we examine the effect of many color orderings on the condition number of the model system preconditioned by incomplete Cholesky factors. We prove the limited result that a particular four-color ordering results in a smaller condition number than a two-color, R/B ordering. We have observed experimentally (see Figure 1) that the convergence continues to improve as the number of colors increases beyond four.

We consider a slightly different four-coloring from that used in the preceding section.

$$(4.30) \quad \begin{array}{cccc} 2 & 4 & 2 & 4 \\ 1 & 3 & 1 & 3 \\ 2 & 4 & 2 & 4 \\ 1 & 3 & 1 & 3 \end{array}$$

Consider the discrete Laplacian matrix  $A$  ordered according to this coloring. The incomplete factorization  $A \approx LL^T$  results in the local operator,  $L_{j,k}$ , acting at the grid point  $(j, k)$  given by

$$(4.31) \quad L_{j,k} = \begin{cases} 1, & (j, k) \text{ color 1} \\ \sqrt{\frac{7}{8}} - \frac{1}{4}(E_y + E_y^{-1}), & (j, k) \text{ color 2} \\ \sqrt{\frac{7}{8}} - \frac{1}{4}(E_x + E_x^{-1}), & (j, k) \text{ color 3} \\ \sqrt{\frac{5}{7}} - \sqrt{\frac{2}{7}}(E_x + E_x^{-1} + E_y + E_y^{-1}), & (j, k) \text{ color 4} \end{cases},$$

where  $E_x$  and  $E_y$  are the shift operators in the  $x$  and  $y$  directions. As observed by Kuo and Chan [8], this representation is accurate only far from the boundary. Therefore, the results derived from equation 4.31 are valid asymptotically.

We note that the Fourier representation of the preconditioned system,  $\hat{L}^{-1}\hat{A}\hat{L}^{-T}$ , is block diagonal, using the basis given in equation 2.3. The block corresponding to the frequency pair  $(\eta, \xi)$  is given by

$$(4.32) \quad \hat{L}_{\xi,\eta}^{-1}\hat{A}_{\xi,\eta}\hat{L}_{\xi,\eta}^{-T} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{-2(-2+a)(2+a)}{7} & \frac{-2ab}{7} & \frac{2b-8a^2b}{7\sqrt{2}\sqrt{5}} \\ 0 & \frac{-2ab}{7} & \frac{-2(-2+b)(2+b)}{7} & \frac{2a-8b^2a}{7\sqrt{2}\sqrt{5}} \\ 0 & \frac{2b-8a^2b}{7\sqrt{2}\sqrt{5}} & \frac{2a-8b^2a}{7\sqrt{2}\sqrt{5}} & \frac{49-12a^2-12b^2}{35} \end{pmatrix},$$

where  $a = \cos \xi \pi h$  and  $b = \cos \eta \pi h$ .

If  $\lambda_{\xi,\eta}$  are the eigenvalues of block matrix in equation 4.32, the condition number of the preconditioned system is given by

$$(4.33) \quad \kappa(L^{-1}AL^{-T}) = \frac{\max_{\xi,\eta} |\lambda_{\xi,\eta}|}{\min_{\xi,\eta} |\lambda_{\xi,\eta}|}.$$

It is evident that the maximum and minimum eigenvalues occur for  $a = b = 0$  and  $(\eta, \xi) = (1, 1)$ , respectively. Thus, we find that

$$(4.34) \quad \max_{\xi, \eta} |\lambda_{\xi, \eta}| = \frac{7}{5},$$

and

$$(4.35) \quad \min_{\xi, \eta} |\lambda_{\xi, \eta}| = \frac{1}{70} \left[ 89 - 44 \cos^2(\pi h) - 16 \cos^4(\pi h) \right] - \frac{1}{70} \left[ 81 + 8 \cos^2(\pi h) - 912 \cos^4(\pi h) + 1408 \cos^6(\pi h) + 256 \cos^8(\pi h) \right]^{\frac{1}{2}}.$$

We compute that  $\min_{\xi, \eta} |\lambda_{\xi, \eta}| = \frac{56}{29}(\pi h)^2 + o(h^2)$ . Thus, we obtain the approximation

$$(4.36) \quad \kappa(L^{-1}AL^{-T}) \approx \frac{29}{40\pi^2 h^2}.$$

If two colors are used, Kuo and Chan [8] prove that the condition number is asymptotically equal to  $\frac{1}{\pi^2 h^2}$ .

In Table 3 our approximations to the eigenvalues are compared to the calculated eigenvalues for several mesh sizes. We observe that the computed values converge to predicted values with increasing mesh size.

TABLE 3  
*Comparison of calculated eigenvalues and predicted eigenvalues for ICCG*

Mesh Size	Approximate Maximum $\lambda$	Calculated Maximum $\lambda$	Approximate Minimum $\lambda$	Calculated Minimum $\lambda$
$4 \times 4$	1.4000	1.1197	0.7623	0.4435
$8 \times 8$	1.4000	1.3219	0.2347	0.2154
$16 \times 16$	1.4000	1.3782	0.0659	0.0644
$32 \times 32$	1.4000	1.3943	0.0175	0.0174

Experimental evidence similar to the graph in §1 supports the conjecture that increasing the number of colors beyond four will continue to improve the convergence rate of ICCG. However, proving this conjecture is more difficult than for the multilevel SOR algorithm. The reason for this difficulty is that the block diagonal approximation is asymptotically accurate only for the low-frequency analysis, not for the high-frequency analysis. Because we can accurately determine only the smallest eigenvalue of the preconditioner, an exact bound for the condition number when more colors are used is not yet possible.

**5. Conclusion.** The convergence of iterative methods when used with multicoloring can be improved if the number of colors is increased. For two specific iterative methods, multilevel SOR and ICCG, we have proven that if the number of colors is increased, then the convergence will be faster. This general result is not

limited to specific colorings schemes; it appears to apply to many different coloring schemes.

Using experimental models, we have also shown that the overall execution time can be decreased when more colors are used. A trade-off exists between accelerating convergence by using more colors and improving parallel efficiency by using fewer colors. This trade off should be examined when implementing iterative methods using multicoloring.

## REFERENCES

- [1] L. M. ADAMS, R. J. LEVEQUE, AND D. M. YOUNG, *Analysis of the SOR iteration for the 9-point Laplacian*, SIAM Journal on Numerical Analysis, 25 (1988), pp. 1156–1180.
- [2] T. F. CHAN AND H. C. ELMAN, *Fourier analysis of iterative methods for elliptic boundary value problems*, SIAM Review, 31 (1989), pp. 20–49.
- [3] H. C. ELMAN AND E. AGRÓN, *Ordering techniques for the preconditioned conjugate gradient method on parallel computers*, Computer Physics Communications, 53 (1989), pp. 253–269.
- [4] L. A. HAGEMAN AND D. M. YOUNG, *Applied Iterative Methods*, Academic Press, New York, 1981.
- [5] D. L. HARRAR II, *Conjugate Gradient Methods for Red/Black Systems on Vector Computers*, Ph.D. thesis, Department of Applied Mathematics, University of Virginia, 1989.
- [6] M. T. JONES AND P. E. PLASSMANN, *Scalable iterative solution of sparse linear systems*, Preprint MCS-P277-1191, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Ill., 1991.
- [7] M. T. JONES AND P. E. PLASSMANN, *A parallel graph coloring heuristic*, SIAM Journal on Scientific and Statistical Computing (to appear).
- [8] C.-C. J. KUO AND T. F. CHAN, *Two-color Fourier analysis of iterative algorithms for elliptic problems with red/black ordering*, SIAM Journal on Scientific and Statistical Computing, 11 (1990), pp. 767–793.
- [9] C.-C. J. KUO AND B. C. LEVY, *A two-level four-color SOR method*, SIAM Journal on Numerical Analysis, 26 (1989), pp. 129–151.
- [10] P. J. LANZKRON, D. J. ROSE, AND D. B. SZYLD, *Convergence of nested classical iterative methods for linear systems*, Numerische Mathematik, 58 (1991), pp. 685–702.
- [11] T. A. MANTEUFFEL, *An incomplete factorization technique for positive definite linear systems*, Mathematics of Computation, 34 (1980), pp. 473–497.
- [12] J. MEIJERINK AND H. VAN DER VORST, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Mathematics of Computation, 31 (1977), pp. 148–162.
- [13] J. M. ORTEGA, *Introduction to Parallel and Vector Solution of Linear Systems*, Plenum, New York, 1988.
- [14] S. V. PARTER, *“Multi-line” iterative methods for elliptic difference equations and fundamental frequencies*, Numerische Mathematik, 3 (1961), pp. 305–319.
- [15] D. M. YOUNG, *Iterative Solution of Large Linear Systems*, Academic Press, New York, 1971.