August 21, 1987 LIDS-P-1625

A Two-level Four-color SOR Method*

C.-C. Jay Kuo†

Bernard C. Levy††

Abstract

A 2-level 4-color SOR method is proposed for the 9-point discretization of the Poisson equation on a square. Instead of examining the Jacobi iteration matrix in the space domain, we consider an equivalent but much simpler 4-color iteration matrix in the frequency domain. A 2-level SOR method is introduced to increase the convergence rate for the frequency-domain iteration matrix. At a first level, the red and orange points, and then the black and green points are treated as groups, and a block SOR iteration is performed on these two groups. At a second level, another SOR iteration is used to decouple values at the red and orange points, and then at the black and green points. The conventional red/black SOR iteration for a 5-point stencil is shown to be a degenerate case of the general 2-level 4-color SOR method. For the case of the 9-point stencil, a closed-form expression for the optimal relaxation parameters ω_b^* and ω_p^* at the two iteration levels is given, and the efficiency of the resulting method is shown both analytically and numerically.

Key Words: Successive Overrelaxation, 9-point discretization, Multicolor SOR, 2-level Iteration.

---

* This work was supported in part by the Army Research Office under grant numbers DAAG29-84-K-0005 and DAAL03-86-K-0171, in part by the Advanced Research Projects Agency monitored by ONR under contract N00014-81-K-0742, and in part by AFOSR Contract F49620-84-C-0004, and was performed when the authors were with the Department of Electrical Engineering and Computer Science at the Massachusetts Institute of Technology.

† Department of Mathematics, University of California, Los Angels, CA 90024.

†† Department of Electrical and Computer Engineering, University of California, Davis, CA 95616.

1. Introduction

The successive overrelaxation (SOR) method introduced in the early 1950's is an effective scheme for accelerating basic relaxation methods such as the Jacobi and Gauss-Seidel iterations [7][12]. The acceleration effect relies on the properties of a special class of matrices known as *consistently ordered* matrices [13] (or *p-cyclic* matrices [11]). By discretizing elliptic PDEs with finite difference schemes, we often obtain sparse matrix equations where the matrix is consistently ordered. Consequently, the SOR method has a wide range of applications.

Three types of approaches, the *pure space* domain, *semi-frequency* domain and *pure frequency* domain approaches, can be used to study the SOR iteration for solving elliptic PDEs. Young's work is a typical example of the pure space domain approach (see Chapters 5 and 6 of [13]). This approach starts from an expression for the SOR iteration in the space domain. Then, under some conditions such as consistent ordering and property A, an argument based on matrix algebra is used to find the relationship between the optimal relaxation parameter for the SOR method and the spectral radii of the Jacobi and SOR iteration matrices.

The approach used in [2], [7], [8], [9] and [10] can be viewed as a semi-frequency domain approach, which adopts the space domain formulation but uses a frequency domain, or Fourier, analysis technique. This approach still starts from a fixed expression for the SOR iteration in the space domain. Then, under the assumption that the PDEs have constant coefficients and are defined on a rectangular domain with Dirichlet or periodic boundary conditions, sinusoidal functions turn out to be eigenfunctions of the discretized system of equations [7][10]. Hence, the system of equations can be

decoupled by using these functions as a basis and, as a consequence, each frequency can be considered separately. This approach, although only rigorous for a restricted class of problems, provides a simple explanation of how the SOR method works.

Another example of semi-frequency domain approach is the analysis presented in Chapter 7 of [13], where the spectral properties of the Jacobi and SOR iteration matrices are studied by performing similarity transformations on these matrices so that they become respectively a diagonal matrix and a matrix in Jordan canonical form. However, since this procedure does not assume any special form of the basis functions for these similarity transformations, the result obtained by this analysis is more general and can be applied to a large class of problems such as space-varying coefficient PDEs on irregular domains.

A common feature of the above two approaches is that an SOR iteration form in the space domain has to be specified a priori. For simple cases such as for a 5-point discretization of the Poisson equation, most reasonable SOR iteration forms lead to an analysis in which the optimal relaxation parameter can be determined in closed form. However, for more complicated cases, such as the 9-point stencil case, it is not easy to specify in advance an iteration form whose analysis will be easy [1][3]. A class of 9-point stencil SOR iteration forms in the space domain was analyzed by Adams, LeVeque and Young [2]. Since the iteration matrices obtained from these forms are not consistently ordered, the traditional SOR theory cannot be applied for determining the optimal relaxation parameter. Hence, Adams et al. used a separation of variables technique to study the eigenvalues and eigenfunctions of the system of equations, and showed that the optimal relaxation parameter can be determined by solving a quartic

equation [2].

In this paper, we study the same problem, i.e., we develop an SOR method for the 9-point discretization of the Poisson equation. However, we use a pure frequency domain approach. This approach makes use of the traditional SOR theory for consistently ordered matrices in the *frequency* domain. We first divide grid points into 4 colors: red, orange,black and green. By assuming that the PDE has constant coefficients and is defined on a square, we can apply Fourier analysis to each color so that a 4-color matrix equation can be obtained in the frequency domain. The 4-color matrix is block diagonal with 4×4 matrix blocks along the diagonal. Each of these blocks relates Fourier components of the 4 colors at a single frequency. If we partition the 4×4 matrix associated to a fixed frequency into four 2×2 blocks, it is consistently ordered with respect to blocks. At a first level, we can use a standard block SOR iteration to accelerate the block Jacobi relaxation. Then, to decouple values of two different colors within the same block, we have to invert a 2×2 matrix. This can be easily accomplished by using a point SOR iteration at a second level. Once the appropriate 2-level SOR iteration form is determined in the frequency domain, it is straightforward to transform it back to the space domain.

This procedure yields a new 2-level 4-color SOR method which is completely different from the single-level SOR method studied in [2]. Suppose that all grid points are partitioned into two groups G_1 and G_2 that contain respectively the red and orange points, and the black and green points. One iteration of the 2-level 4-color SOR method consists of the following four steps.

Step 1: The first half part of a block SOR iteration between G_1 and G_2 gives an intermediate function defined at points of G_1.

Step 2: With this intermediate function as driving function, several (usually two) point SOR iterations are performed between red and orange points within G_1 to obtain an updated PDE solution at points of G_1.

Step 3: The second half part of a block SOR iteration between G_1 and G_2 yields an intermediate function defined at points of G_2.

Step 4: With this intermediate function as driving function, several point SOR iterations are performed between black and green points within G_2 to obtain an updated solution at points of G_2.

In the above algorithm, steps 1 and 3 constitute a complete block SOR iteration between groups G_1 and G_2 and define a first level of iteration. Steps 2 and 4 individually consist of several point SOR iteration operations and correspond to a second level of iteration. The values obtained in steps 1 and 3 are used as driving functions in steps 2 and 4 respectively. This computational algorithm will be detailed in Section 3. The optimal relaxation parameters ω_b^* and ω_p^* at the two iteration levels can be expressed in closed-form. The 2-level SOR method is easy to implement, and its spectral radius is of the form $1 - Ch$, where C is a constant comparable to the one obtained in [2].

The paper is organized as follows. In Section 2, we use a simple 1-D 2-color SOR method to demonstrate our pure frequency domain approach. Section 3 describes the main result of this paper, i.e., the 2-D 2-level 4-color SOR algorithm for a 9-point discretization of the Poisson equation. Then, in Section 4, we show that the conventional 2-D single-level 2-color SOR method for the 5-point stencil case is a degenerate case of the general 2-level 4-color scheme. Closed-form formulas for the optimal relaxation parameters ω_b^* and ω_p^* corresponding to the 2 iteration levels are obtained in Section 5, where the convergence rate of the 2-level SOR method is also analyzed.

Finally, some numerical results are presented in Section 6.

2. 1-D 2-color SOR method

In this section, we consider a simple 1-D model problem and show how the 2-color SOR method can be derived from the Jacobi iteration method by first transforming the problem to the frequency domain and then introducing the relaxation parameter ω inside the frequency-domain iteration matrix. Although the final result is well known, the approach we are taking is new and provides some new insight. The same approach will be used to develop a 2-level iteration method in the next section.

2.1 Problem Formulation

Consider the discrete 1-D Poisson equation on [0,1] with grid spacing h

$$\frac{1}{h^2}\left(u_{j-1} - 2u_j + u_{j+1}\right) = f_j \quad , \quad j = 1, 2, \cdots, N-1 \, ,$$

where u_0, u_N are given, and $N = \frac{1}{h}$. Suppose we divide the problem domain into red and black points corresponding respectively to points with even and odd indices. With this partitioning, the Jacobi iteration method takes the form

$$u_j^{n+1} = \frac{1}{2}\left(u_{j-1}^n + u_{j+1}^n - 2h^2 f_j\right) \quad j \text{ even} \, ,$$

$$u_j^{n+1} = \frac{1}{2}\left(u_{j-1}^n + u_{j+1}^n - 2h^2 f_j\right) \quad j \text{ odd} \, .$$

Denote the exact solution by $\bar{u}_j$ and define the error as $e_j^n = u_j^n - \bar{u}_j$. Then, the error equations can be written as

$$\begin{aligned} e_j^{n+1} &= \frac{1}{2}\left(e_{j-1}^n + e_{j+1}^n\right) \qquad j \text{ even} \, , \\ e_j^{n+1} &= \frac{1}{2}\left(e_{j-1}^n + e_{j+1}^n\right) \qquad j \text{ odd} \, , \end{aligned} \tag{2.1}$$

with $e_0 = e_N = 0$.

Since (2.1) is a system of linear constant-coefficient equations with homogeneous boundary conditions, the eigenfunctions of this system are given by $\sin(\xi\pi jh)$, where $\xi = 1, 2, \cdots, (N-1)$. These functions form a basis, so that

$$\begin{aligned} r_j^n &= \sum_{\xi=1}^{N-1} \hat{r}_\xi^n \sin(\xi\pi jh) \quad 1 \leqslant j \leqslant N-1, \\ b_j^n &= \sum_{\xi=1}^{N-1} \hat{b}_\xi^n \sin(\xi\pi jh) \quad 1 \leqslant j \leqslant N-1, \end{aligned} \tag{2.2}$$

where the coefficients $\hat{r}_\xi^n$ and $\hat{b}_\xi^n$ are chosen such that

$$\begin{aligned} r_j^n &= e_j^n \qquad j \text{ even}, \\ b_j^n &= e_j^n \qquad j \text{ odd}. \end{aligned} \tag{2.3}$$

In other words, r_j^n and b_j^n are two sequences which coincide with the errors at red and black points respectively. They can be viewed as interpolations of the errors at the red and black points to all grid points. Note that there are $2(N-1)$ undetermined coefficients in (2.2) and only $N-1$ constraints in (2.3). Since (2.2) and (2.3) form a underdetermined system of equation, there are many ways to choose $\hat{r}_\xi^n$ and $\hat{b}_\xi^n$. However, the actual values of these coefficients are not important. We are primarily concerned with how they evolve as the iteration proceeds.

Consider the error dynamics relating r_j and b_j,

$$\begin{aligned} r_j^{n+1} &= \frac{1}{2}\left(b_{j-1}^n + b_{j+1}^n\right) \qquad 1 \leqslant j \leqslant N-1, \\ b_j^{n+1} &= \frac{1}{2}\left(r_{j-1}^n + r_{j+1}^n\right) \qquad 1 \leqslant j \leqslant N-1. \end{aligned} \tag{2.4}$$

Although (2.4) contains more information than (2.1), all the information contained in (2.1) is preserved by (2.4) and the dynamic behavior of (2.1) can be obtained by studying the dynamic behavior of (2.4). Conceptually, (2.4) is easier to analyze than (2.1) since it is a spatially invariant system for both red and black colors.

By substituting (2.2) inside (2.4), for $\xi = 1, 2, ..., N-1$, we have

$$\begin{bmatrix} \hat{r}_\xi^{n+1} \\ \hat{b}_\xi^{n+1} \end{bmatrix} = B(\xi) \begin{bmatrix} \hat{r}_\xi^{n} \\ \hat{b}_\xi^{n} \end{bmatrix}, \tag{2.5}$$

where

$$B(\xi) = \begin{bmatrix} 0 & \cos(\xi\pi h) \\ \cos(\xi\pi h) & 0 \end{bmatrix} \tag{2.6}$$

is called the *Jacobi iteration matrix* for the frequency $\xi\pi$, which has two eigenvalues

$$\mu_\xi = \pm \cos(\xi\pi h).$$

Intuitively speaking, we use the fact that the sinusoidal functions are eigenfunctions of the linear system (2.4) so that, by changing the coordinate from the space domain to the frequency domain, we are able to decompose the loosely coupled system (2.4) into a decoupled system which is a block diagonal matrix containing many 2×2 matrices along the diagonal.

Since the spectral radius of $B(\xi)$ is less than 1 for any ξ, the iteration (2.5) converges. Consequently, the asymptotic values $\hat{r}_\xi^\infty$ and $\hat{b}_\xi^\infty$ obtained by this iteration procedure are $\hat{r}_\xi^\infty = \hat{b}_\xi^\infty = 0$, and (2.5) can be viewed as obtained by solving the linear system

$$A(\xi) \begin{bmatrix} \hat{r}_\xi^\infty \\ \hat{b}_\xi^\infty \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \text{with} \quad A(\xi) = \begin{bmatrix} 1 & -\cos(\xi\pi h) \\ -\cos(\xi\pi h) & 1 \end{bmatrix},$$

by the Jacobi iteration in the frequency domain. In order to increase the convergence rate of (2.5), we have to reduce the spectral radius of $B(\xi)$.

2.2 Point SOR Iteration

The key idea of this paper is that instead of considering the SOR method for the large matrix corresponding to (2.1), we can study the SOR scheme for each small 2×2

matrix given by (2.6) separately and, then, seek the best SOR scheme for all of them. Once the SOR scheme is obtained in the frequency domain, we transform the problem back to the space domain so that the corresponding spatial SOR iteration can be determined.

It is important to observe in this context that $A(\xi)$ and $B(\xi)$ are *consistently ordered*. Since the SOR method was originally developed to accelerate the convergence rate of consistently ordered matrices, the SOR method can be applied directly to the iteration (2.5). The definitions of consistent ordering, and the details of the SOR theory are all presented in [11] and [13].

Since this is a standard procedure, we only summarize the result here. Let

$$A(\xi) = I - L(\xi) - U(\xi)$$

where $L(\xi)$ and $U(\xi)$ are lower and upper triangular matrices respectively. Then, for a fixed frequency $\xi\pi$, the Jacobi iteration matrix is

$$B(\xi) = L(\xi) + U(\xi)$$

and the SOR iteration matrix associated with the frequency $\xi\pi$ is

$$G_\omega(\xi) = (I - \omega L(\xi))^{-1} \{ (1 - \omega) I + \omega U(\xi) \} . \qquad (2.7)$$

In addition, the eigenvalues λ_ξ of $G_\omega(\xi)$ and the eigenvalues μ_ξ of $B(\xi)$ are related by [11, p.106]

$$(\lambda_\xi + \omega_\xi - 1)^2 = \lambda_\xi \omega_\xi^2 \mu_\xi^2 .$$

Hence,

$$\lambda_\xi = \left(\frac{\omega_\xi \mu_\xi \pm \sqrt{\Delta}}{2} \right)^2 \quad \text{where} \quad \Delta = \omega_\xi^2 \mu_\xi^2 - 4 (\omega_\xi - 1) ,$$

and the spectral radius of (2.7) is

$$\rho_\xi = \begin{cases} \left(\dfrac{|\omega_\xi \mu_\xi| \pm \sqrt{\Delta}}{2} \right)^2 & \text{if } \Delta > 0 \\ \omega_\xi - 1 & \text{if } \Delta \leqslant 0 \end{cases} .$$

The above quantity can be minimized for all ξ by choosing

$$\omega^* = \frac{2}{1 + [1-\mu_{\max}^2]^{1/2}} \quad \text{where} \quad \mu_{\max} = \max_{1 \leqslant \xi \leqslant N-1} |\mu_\xi| = \cos(\pi h) , \tag{2.8}$$

and the resulting spectral radius is

$$\rho^* = \omega^* - 1 \approx 1 - 2 \sin(\pi h) = 1 - 2\pi h .$$

In particular, since the SOR method is applied to $A(\xi)$ partitioned with 1×1 diagonal submatrices, we call it the *point* SOR method.

The remaining problem is to transform the SOR iteration matrix (2.7) back to the space domain. By using the correspondence,

$$\cos(\xi \pi l h) = \frac{1}{2}(e^{i\xi\pi l h} + e^{-i\xi\pi l h}) \quad \longleftrightarrow \quad \frac{1}{2}(E^l + E^{-l}) \quad l = 1, 2, \cdots ,$$

where E^l and E^{-l} are the l-th order forward and backward shift operators defined as $E^l u_j = u_{j+l}$ and $E^{-l} u_j = u_{j-l}$, we find that the SOR iteration for r_j^n and b_j^n becomes

$$\begin{aligned} r_j^{n+1} &= (1 - \omega^*) r_j^n + \frac{\omega^*}{2} (b_{j-1}^n + b_{j+1}^n) \\ b_j^{n+1} &= (1 - \omega^*) b_j^n + \frac{\omega^*}{2} (r_{j-1}^{n+1} + r_{j+1}^{n+1}) \end{aligned} \tag{2.9}$$

It is straightforward to reconstruct the SOR iteration from (2.9), i.e.

$$u_j^{n+1} = (1 - \omega^*) u_j^n + \frac{\omega^*}{2} (u_{j-1}^n + u_{j+1}^n - h^2 f_j) \quad j \text{ even} ,$$

$$u_j^{n+1} = (1 - \omega^*) u_j^n + \frac{\omega^*}{2} (u_{j-1}^{n+1} + u_{j+1}^{n+1} - h^2 f_j) \quad j \text{ odd} ,$$

which is consistent with the conventional SOR method with red/black partitioning.

3. 2-D 2-Level 4-color SOR method

3.1 Problem formulation

The 1-D 2-color SOR scheme discussed in the previous section can be naturally generalized to the 2-D case by using 4 colors.

Consider the following discretized system with uniform grid spacing h,

$$\frac{1}{h^2}\{ q_1(u_{j+1,k}+u_{j-1,k})+q_2(u_{j,k+1}+u_{j,k-1})+q_3(u_{j+1,k+1}+u_{j+1,k-1} +u_{j-1,k+1}+u_{j-1,k-1}) - q\ u_{j,k}\} = f_{j,k} \quad j,k = 1, 2, \cdots, N-1, \tag{3.1}$$

where

$$q = 2q_1 + 2q_2 + 4q_3,$$

and $N = \frac{1}{h}$, and where q_1, q_2, and q_3 are nonnegative and not all zeros. It is also assumed that values at all boundary points are given. The system (3.1) can be viewed as obtained from a 5-point or 9-point stencil discretization [5] of the equation

$$q'_1 \frac{\partial^2 u(x,y)}{\partial x^2} + q'_2 \frac{\partial^2 u(x,y)}{\partial y^2} = f(x,y) \quad \text{where } q'_1, q'_2 > 0, \tag{3.2}$$

on the unit square $[0,1]^2$ with Dirichlet boundary conditions. In particular, when $q'_1 = q'_2$, (3.2) becomes the Poisson equation. This section presents a frequency domain approach for the design of a 2-level 4-color SOR method to solve (3.1). Several concrete examples will then be examined in Sections 4-6.

We can divide the grid points into four groups, say, red, black, green, and orange. A grid point is red if both j and k are even, black if j is odd and k is even, green if j is even and k is odd, and orange if both j and k are odd, as shown in Figure 1. Following the procedure described in the previous section, to understand the error dynamics of the error associated to the Jacobi iteration for the system (3.1), we

examine the dynamics of the four 2-D sequences

$$r^n_{j,k} = \sum_{\xi=1}^{N-1}\sum_{\eta=1}^{N-1} \hat{r}^n_{\xi,\eta} \sin(\xi\pi jh)\sin(\eta\pi kh) \quad 1 \leqslant j,k \leqslant N-1,$$

$$b^n_{j,k} = \sum_{\xi=1}^{N-1}\sum_{\eta=1}^{N-1} \hat{b}^n_{\xi,\eta} \sin(\xi\pi jh)\sin(\eta\pi kh) \quad 1 \leqslant j,k \leqslant N-1,$$

$$g^n_{j,k} = \sum_{\xi=1}^{N-1}\sum_{\eta=1}^{N-1} \hat{g}^n_{\xi,\eta} \sin(\xi\pi jh)\sin(\eta\pi kh) \quad 1 \leqslant j,k \leqslant N-1,$$

$$o^n_{j,k} = \sum_{\xi=1}^{N-1}\sum_{\eta=1}^{N-1} \hat{o}^n_{\xi,\eta} \sin(\xi\pi jh)\sin(\eta\pi kh) \quad 1 \leqslant j,k \leqslant N-1,$$

where the coefficients $\hat{r}^n_{\xi,\eta}, \hat{b}^n_{\xi,\eta}, \hat{g}^n_{\xi,\eta}, \hat{o}^n_{\xi,\eta}$ are chosen such that

$$r^n_{j,k} = e^n_{j,k} \quad j \text{ even } k \text{ even}, \quad b^n_{j,k} = e^n_{j,k} \quad j \text{ odd } k \text{ even},$$

$$g^n_{j,k} = e^n_{j,k} \quad j \text{ even } k \text{ odd}, \quad o^n_{j,k} = e^n_{j,k} \quad j \text{ odd } k \text{ odd},$$

where $e^n_{j,k} = u^n_{j,k} - \bar{u}_{j,k}$ is the n th iteration error at grid point (jh, kh).

As shown before, we can transform the Jacobi iteration for $r^n_{j,k}$, $b^n_{j,k}$, $g^n_{j,k}$ and $o^n_{j,k}$, or equivalently for the errors in the space domain at the red, black, green, and orange points, into an equivalent set of iterations for the Fourier coefficients $\hat{r}^n_{\xi,\eta}$, $\hat{b}^n_{\xi,\eta}$, $\hat{g}^n_{\xi,\eta}$ and $\hat{o}^n_{\xi,\eta}$ in the frequency domain. With respect to the frequency domain vector $(\hat{r}^n_{\xi,\eta}, \hat{o}^n_{\xi,\eta}, \hat{b}^n_{\xi,\eta}, \hat{g}^n_{\xi,\eta})^T$, these iterations can be viewed as solving the system

$$A(\xi,\eta) \begin{pmatrix} \hat{r}^\infty_{\xi,\eta} \\ \hat{o}^\infty_{\xi,\eta} \\ \hat{b}^\infty_{\xi,\eta} \\ \hat{g}^\infty_{\xi,\eta} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{3.3a}$$

where $A(\xi,\eta)$ is the frequency domain *coefficient* matrix for the frequency $(\xi\pi, \eta\pi)$ and has the form

$$A(\xi,\eta) = \begin{bmatrix} 1 & -\alpha_3 & -\alpha_1 & -\alpha_2 \\ -\alpha_3 & 1 & -\alpha_2 & -\alpha_1 \\ -\alpha_1 & -\alpha_2 & 1 & -\alpha_3 \\ -\alpha_2 & -\alpha_1 & -\alpha_3 & 1 \end{bmatrix}, \tag{3.3b}$$

where

$$\alpha_1 = \frac{2q_1\cos(\xi\pi h)}{q}, \quad \alpha_2 = \frac{2q_2\cos(\eta\pi h)}{q}, \quad \alpha_3 = \frac{4q_3\cos(\xi\pi h)\cos(\eta\pi h)}{q}. \tag{3.4}$$

Notice that the coefficient matrix $A(\xi,\eta)$ in (3.3) is symmetric and diagonally dominant with positive diagonal elements. The application of the Jacobi, block Jacobi, SOR, and block SOR iterations with $0 < \omega < 2$ to the system of equations (3.3) is expected to converge [11][13].

3.2 Block SOR iteration

The matrix $A(\xi,\eta)$ partitioned such that its diagonal submatrices are all 1×1 matrices is *not* a consistently ordered matrix. However, if $A(\xi,\eta)$ is partitioned with 2×2 block diagonal submatrices, it is consistently ordered with respect to blocks. Hence, a *block* SOR iteration can be applied to $A(\xi,\eta)$ with this kind of partitioning.

The matrix A can be written as

$$A(\xi,\eta) = D(\xi,\eta) - E(\xi,\eta) - F(\xi,\eta)$$

where

$$D(\xi,\eta) = \begin{bmatrix} 1 & -\alpha_3 & 0 & 0 \\ -\alpha_3 & 1 & 0 & 0 \\ 0 & 0 & 1 & -\alpha_3 \\ 0 & 0 & -\alpha_3 & 1 \end{bmatrix}, \quad E(\xi,\eta) = \begin{bmatrix} 0 & 0 & \alpha_1 & \alpha_2 \\ 0 & 0 & \alpha_2 & \alpha_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

and $F(\xi,\eta) = E^T(\xi,\eta)$. In addition, we can define $L(\xi,\eta) \equiv D^{-1}(\xi,\eta)E(\xi,\eta)$ and $U(\xi,\eta) \equiv D^{-1}(\xi,\eta)F(\xi,\eta)$, i.e.,

$$L(\xi,\eta) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \beta_1 & \beta_2 & 0 & 0 \\ \beta_2 & \beta_1 & 0 & 0 \end{bmatrix}, \quad U(\xi,\eta) = \begin{bmatrix} 0 & 0 & \beta_1 & \beta_2 \\ 0 & 0 & \beta_2 & \beta_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

where

$$\beta_1 = \frac{\alpha_1 + \alpha_2\alpha_3}{1-\alpha_3^2}, \qquad \beta_2 = \frac{\alpha_2 + \alpha_1\alpha_3}{1-\alpha_3^2}.$$

Then, the block Jacobi iteration matrix is

$$B(\xi,\eta) = L(\xi,\eta) + U(\xi,\eta) \ , \tag{3.5}$$

and the corresponding block SOR iteration matrix is

$$G_\omega(\xi,\eta) = (I - \omega L(\xi,\eta))^{-1} \{ (1-\omega) I + \omega U(\xi,\eta) \} . \tag{3.6}$$

It is easy to find that the eigenvalues of $B(\xi,\eta)$ are double roots at

$$\mu_{\xi,\eta} = \beta_1 \pm \beta_2 = \frac{\alpha_1+\alpha_2}{1-\alpha_3}, \frac{\alpha_1-\alpha_2}{1+\alpha_3} . \tag{3.7}$$

In addition, the eigenvalues $\mu_{\xi,\eta}$ of the Jacobi iteration matrix $B(\xi,\eta)$ and the eigenvalues $\lambda_{\xi,\eta}$ of the SOR iteration matrix $G_\omega(\xi,\eta)$ are related by

$$(\lambda_{\xi,\eta} + \omega_{\xi,\eta} - 1)^2 = \lambda_{\xi,\eta}\, \omega_{\xi,\eta}^2\, \mu_{\xi,\eta}^2 .$$

Hence, if we proceed as in the 1-D case, except for a change of subscript from the 1-D index ξ to the 2-D index (ξ,η), we find that

$$\lambda_{\xi,\eta} = \left(\frac{\omega_{\xi,\eta}\mu_{\xi,\eta} \pm \sqrt{\Delta}}{2} \right)^2 \quad \text{where} \quad \Delta = \omega_{\xi,\eta}^2\mu_{\xi,\eta}^2 - 4 (\omega_{\xi,\eta} - 1)$$

and the spectral radius of $G_\omega(\xi,\eta)$ is

$$\rho_{\xi,\eta} = \begin{cases} \left(\dfrac{|\omega_{\xi,\eta}\mu_{\xi,\eta}| \pm \sqrt{\Delta}}{2} \right)^2 & \text{if } \Delta > 0 \\ \omega_{\xi,\eta} - 1 & \text{if } \Delta \leqslant 0 \end{cases} .$$

The above quantity is minimized for all ξ and η by choosing the following optimal relaxation parameter

$$\omega^* = \frac{2}{1 + [1-\mu_{\max}^2]^{1/2}} \quad \text{where} \quad \mu_{\max} = \max_{1 \leq \xi,\eta \leq N-1} |\mu_{\xi,\eta}| \ , \tag{3.8}$$

and the spectral radius of the corresponding SOR matrix is

$$\rho^* = \omega^* - 1 \ .$$

3.3 2-level SOR iteration

Suppose that one of the coefficients q_1, q_2 or q_3 is zero, or equivalently, that one of α_1, α_2, or α_3 is zero for all (ξ,η). Then, the 4-color block SOR method described above reduces to an equivalent 2-color SOR method, which corresponds to a degenerate case that will be discussed in Section 4.

For the moment, consider the nondegenerate case where q_1, q_2, and q_3 are all strictly positive. In this case, the *pure* frequency-domain block SOR method given by (3.6) cannot be successfully transformed back to the space domain. We can rewrite (3.6) as

$$G_\omega(\xi,\eta) = (D(\xi,\eta) - \omega E(\xi,\eta))^{-1} \{ (1 - \omega) D(\xi,\eta) + \omega F(\xi,\eta) \} ,$$

and the corresponding space domain equation associated to the optimal block relaxation parameter ω^* becomes

$$\begin{aligned}
r_{j,k}^{n,+1} - \alpha_3^S o_{j,k}^{n,+1} &= (1 - \omega^*) (\ r_{j,k}^{n} - \alpha_3^S o_{j,k}^{n}) + \omega^* (\alpha_1^S b_{j,k}^{n} + \alpha_2^S g_{j,k}^{n}), \\
- \alpha_3^S r_{j,k}^{n,+1} + o_{j,k}^{n,+1} &= (1 - \omega^*) (- \alpha_3^S r_{j,k}^{n} + o_{j,k}^{n}) + \omega^* (\alpha_2^S b_{j,k}^{n} + \alpha_1^S g_{j,k}^{n}), \\
b_{j,k}^{n,+1} - \alpha_3^S g_{j,k}^{n,+1} &= (1 - \omega^*) (\ b_{j,k}^{n} - \alpha_3^S g_{j,k}^{n}) + \omega^* (\alpha_1^S r_{j,k}^{n,+1} + \alpha_2^S o_{j,k}^{n,+1}), \\
- \alpha_3^S b_{j,k}^{n,+1} + g_{j,k}^{n,+1} &= (1 - \omega^*) (- \alpha_3^S b_{j,k}^{n} + g_{j,k}^{n}) + \omega^* (\alpha_2^S r_{j,k}^{n,+1} + \alpha_1^S o_{j,k}^{n,+1}),
\end{aligned} \tag{3.9}$$

where α_1^S, α_2^S, and α_3^S are space domain operators corresponding respectively to α_1, α_2, and α_3 respectively, i.e.

$$\alpha_1^S = \frac{q_1}{q}\,(E_1 + E_1^{-1}), \qquad \alpha_2^S = \frac{q_2}{q}\,(E_2 + E_2^{-1}),$$

$$\alpha_3^S = \frac{q_3}{q}\,(E_1 + E_1^{-1})(E_2 + E_2^{-1}).$$

At the $n+1$th iteration, the values of the solution at the red and orange points and at the black and green points are coupled together as indicated by the left-hand side of (3.9)

We can divide (3.9) into two sets of equations: the first two and the last two. Within each set, for example the first two equations, instead of solving for $r_{j,k}^{n+1}$ and $o_{j,k}^{n+1}$ directly, we can apply a point SOR scheme to these two equations and compute $r_{j,k}^{n+1}$ and $o_{j,k}^{n+1}$ iteratively. As a consequence, we obtain a 2-level SOR method, which is described in Table 1 and that can be explained as follows. Let us treat all red and orange points as one group G_1, and all black and green points as the other group G_2. Then, this 2-level 4-color SOR method includes 4 steps.

Step 1: Compute the intermediate function $ff_{j,k}$ defined in Table 1 at points of G_1 by performing the first half of a block SOR iteration between G_1 and G_2.

Step 2: Perform M point SOR iterations between red and orange points M times with $ff_{j,k}$ as driving function.

Step 3: Compute the function $ff_{j,k}$ at points of G_2 by performing the second half of a block SOR iteration between G_1 and G_2.

Step 4: Perform M point SOR iterations between black and green points with $ff_{j,k}$ as driving function.

In the above procedure, steps 1 and 3 form a complete block SOR iteration between groups G_1 and G_2 which is the first level of iteration. The output of this iteration, denoted by $ff_{j,k}^{n+1}$, is used as driving function for the second level iteration. Each step

in steps 2 and 4 consists of M point SOR iterations, which is the second level of iteration used to decouple the values of the solution at red and orange points, or at black and green points.

for $n = 0, 1, 2, \cdots$

for $j+k$ even /G_1 : red & orange/

$ff_{j,k}^{n+1} := (1-\omega_b)(u_{j,k}^n - \alpha_3^S u_{j,k}^n) + \omega_b(\alpha_1^S u_{j,k}^n + \alpha_2^S u_{j,k}^n - h^2 f_{j,k})$ (step 1)

$v_{j,k}^0 := u_{j,k}^n$

for $m = 0, 1, 2, \ldots, M-1$ (step 2)

for k even /red/

$v_{j,k}^{m+1} := (1-\omega_p)v_{j,k}^m + \omega_p(\alpha_3^S v_{j,k}^m + ff_{j,k}^{n+1})$

for k odd /orange/

$v_{j,k}^{m+1} := (1-\omega_p)v_{j,k}^m + \omega_p(\alpha_3^S v_{j,k}^{m+1} + ff_{j,k}^{n+1})$

$u_{j,k}^{n+1} := v_{j,k}^M$

for $j+k$ odd /G_2 : black & green/

$ff_{j,k}^{n+1} := (1-\omega_b)(u_{j,k}^n - \alpha_3^S u_{j,k}^n) + \omega_b(\alpha_1^S u_{j,k}^{n+1} + \alpha_2^S u_{j,k}^{n+1} - h^2 f_{j,k})$ (step 3)

$v_{j,k}^0 := u_{j,k}^n$

for $m = 0, 1, 2, \ldots, M-1$ (step 4)

for k even /black/

$v_{j,k}^{m+1} := (1-\omega_p)v_{j,k}^m + \omega_p(\alpha_3^S v_{j,k}^m + ff_{j,k}^{n+1})$

for k odd /green/

$v_{j,k}^{m+1} := (1-\omega_p)v_{j,k}^m + \omega_p(\alpha_3^S v_{j,k}^{m+1} + ff_{j,k}^{n+1})$

$u_{j,k}^{n+1} := v_{j,k}^M$

Table 1: A 2-level 4-color SOR Method

A data flow diagram which illustrates how grid points exchange information with their neighboring points at each step of one iteration is shown in Figure 2. For simplicity, only one point SOR iteration in steps 2 and 4 is illustrated in this data flow diagram. Note that we use different relaxation parameters at different levels, i.e., we

use respectively ω_b and ω_p for the block and point SOR iterations.

It is a well known result that both the block and point SOR iterations applied to a symmetric positive definite matrix converge if and only if their relaxation parameters are between 0 and 2 [11]. Hence, the convergence of the 2-level SOR scheme can be achieved by first selecting

$$0 < \omega_p < 2\,, \quad M \text{ sufficiently large}\,, \tag{3.10a}$$

where M denotes the total number of point SOR iterations performed at the second level, so that the point SOR iteration converges inside each block SOR iteration. Under condition (3.10a), a 2-level SOR iteration is not different from a single-level block SOR iteration. Therefore, by imposing the additional constraint,

$$0 < \omega_b < 2\,, \tag{3.10b}$$

the 2-level SOR method is guaranteed to converge. In Section 5, we will discuss how to select the number M and optimal relaxation parameters ω_p^* and ω_b^* to maximize the convergence rate of the 2-level SOR method.

3.4 Rederivation of Adams et al's SOR method

It is possible to derive the SOR method of Adams, LeVeque and Young [2] directly from the frequency domain matrix equation (3.3). To do so, rewrite the coefficient matrix $A(\xi,\eta)$ as

$$A(\xi,\eta) = \tilde{D}(\xi,\eta) - \tilde{E}(\xi,\eta) - \tilde{F}(\xi,\eta)$$

where

$$\tilde{D}(\xi,\eta) = I = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \tilde{E}(\xi,\eta) = \begin{bmatrix} 0 & \alpha_3 & \alpha_1 & \alpha_2 \\ 0 & 0 & \alpha_2 & \alpha_1 \\ 0 & 0 & 0 & \alpha_3 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

and $\tilde{F}(\xi,\eta) = \tilde{E}^T(\xi,\eta)$. In the frequency domain, we can then consider an SOR iteration of the form

$$\tilde{G}_\omega(\xi,\eta) = (I - \omega \tilde{F}(\xi,\eta))^{-1} \{ (1-\omega) I + \omega \tilde{E}(\xi,\eta) \} . \tag{3.11}$$

In the space domain, (3.11) corresponds to Adams et al.'s SOR method with R/O/B/G ordering, which can be written as

$$\begin{aligned}
r_{j,k}^{n,+1} &= (1-\omega) r_{j,k}^{n} + \omega (\alpha_1^S b_{j,k}^{n} + \alpha_2^S g_{j,k}^{n} + \alpha_3^S o_{j,k}^{n}) , \\
o_{j,k}^{n,+1} &= (1-\omega) o_{j,k}^{n} + \omega (\alpha_1^S g_{j,k}^{n} + \alpha_2^S b_{j,k}^{n} + \alpha_3^S r_{j,k}^{n,+1}) , \\
b_{j,k}^{n,+1} &= (1-\omega) b_{j,k}^{n} + \omega (\alpha_1^S r_{j,k}^{n,+1} + \alpha_2^S o_{j,k}^{n,+1} + \alpha_3^S g_{j,k}^{n}) , \\
g_{j,k}^{n,+1} &= (1-\omega) g_{j,k}^{n} + \omega (\alpha_1^S o_{j,k}^{n,+1} + \alpha_2^S r_{j,k}^{n,+1} + \alpha_3^S b_{j,k}^{n,+1}) .
\end{aligned} \tag{3.12}$$

If $\tilde{\lambda}_{\xi,\eta}$ is an eigenvalue of $\tilde{G}_\omega(\xi,\eta)$, we have

$$| \tilde{G}_\omega(\xi,\eta) - \tilde{\lambda}_{\xi,\eta} I | = 0 , \tag{3.13}$$

which is a quartic equation of the variable $\tilde{\lambda}_{\xi,\eta}$. In [2], Adams et al. derived a quartic equation in terms of the variable $\gamma = (\tilde{\lambda}_{\xi,\eta})^{1/2}$ and showed that if

$$\gamma^4 + c_1\gamma^3 + c_2\gamma^2 + c_3\gamma + c_4 = 0 , \tag{3.14a}$$

is the quartic equation for the frequency $(\xi\pi,\eta\pi)$, where c_i, $1 \leqslant i \leqslant 4$ are functions of ω, ξ and η, then

$$\gamma^4 - c_1\gamma^3 + c_2\gamma^2 - c_3\gamma + c_4 = 0 \tag{3.14b}$$

is the quartic equation for the frequency $(\xi\pi,(N-\eta)\pi)$. It turns out that the quartic equation (3.13) obtained by our approach is equal to

$$(\gamma^4 + c_1\gamma^3 + c_2\gamma^2 + c_3\gamma + c_4) (\gamma^4 - c_1\gamma^3 + c_2\gamma^2 - c_3\gamma + c_4) = 0 .$$

In other words, the quartic equations (3.13) and (3.14) contain the same amount of information. From (3.13) or (3.14), the optimal relaxation parameter ω^* has to be selected so that the maximum value of $\rho[\tilde{G}_\omega(\xi,\eta)]$ is minimized over all ξ and η. For

the details of this procedure, we refer to [2].

The major advantage of deriving SOR methods directly from the frequency domain coefficient matrix $A(\xi,\eta)$ is that this procedure does not require the knowledge of the eigenvectors of the SOR iteration matrices such as $G_\omega(\xi,\eta)$ in (3.6) and $\tilde{G}_\omega(\xi,\eta)$ in (3.11) for the determination of the optimal relaxation parameters and the corresponding spectral radii. We only have to know the eigenvectors associated to the scalar operators α_1^S, α_2^S and α_3^S which describe the coupling between grid points of different colors. Consequently, the derivation is usually simpler. In addition, if the frequency domain coefficient matrix is block consistently ordered, the standard SOR theory can be applied separately at the block and point levels as shown above and the determination of the optimal block and point relaxation parameters become straightforward.

However, our pure frequency domain approach has several limitations. Sometimes, eigenvectors provide valuable information for understanding the convergence property of an SOR iteration scheme. For example, for the SOR method (3.12), it was found that the eigenvector associated with the spectral radius is highly oscillatory. Therefore, the observed convergence rate for a test problem with a smooth initial error is faster than the predicted convergence rate [2]. Since the eigenvectors for the SOR iteration matrix cannot be found by our approach, this phenomenon cannot be appropriately explained. In addition, our approach does not apply to the SOR method with natural ordering. Even for different coloring schemes such as the one shown in Figure 4, for which we have

$$\alpha_1^S = \frac{1}{q}[\, q_1 E_1 + q_3(\, E_1^{-1}E_2 + E_1^{-1}E_2^{-1}\,)]\,, \quad \alpha_2^S = \frac{q_2}{q}\,(\, E_2 + E_2^{-1}\,)\,,$$

$$\alpha_3^S = \frac{1}{q}[\, q_1 E_1^{-1} + q_3(\, E_1 E_2 + E_1 E_2^{-1}\,)]\,,$$

and where (3.1) can be rewritten in terms of this choice of α_1^S, α_2^S and α_3^S, it is not clear that there exists a frequency domain equation corresponding to (3.3). The difficulty is due to the fact that $\sin(\xi\pi jh)\sin(\eta\pi kh)$ is not an eigenvector of the operators α_1^S and α_3^S any longer. In fact, the results of our paper rely exclusively on the fact that α_1^S, α_2^S and α_3^S admit $\sin(\xi\pi jh)\sin(\eta\pi kh)$ as a common set of eigenvectors, and this requirement is probably the most serious limitation of our approach. Note however that the coloring scheme of Figure 4 is asymmetric, and is therefore less natural than the one that has been considered in this paper.

4. Degenerate case: 5-point stencils

In this section, we show that the traditional single-level 2-color SOR method for a 5-point stencil is in fact a degenerate case of the general 2-level 4-color SOR method described in Table 1. The following discussion also gives us more insight into the 2-level SOR algorithm.

4.1 Standard 5-point stencil

The standard 5-point stencil discretization of the Poisson equation is

$$\frac{1}{h^2}\left(u_{j+1,k} + u_{j-1,k} + u_{j,k+1} + u_{j,k-1} - 4\,u_{j,k} \right) = f_{j,k} ,$$

which is a special case of (3.1) with

$$q_1 = 1 , \quad q_2 = 1 , \quad q_3 = 0 , \quad \text{and} \quad q = 4 .$$

Hence, we have

$$\alpha_1 = \frac{\cos(\xi \pi h)}{2} , \quad \alpha_2 = \frac{\cos(\eta \pi h)}{2} , \quad \alpha_3 = 0 ,$$

and

$$\alpha_1^S = \frac{E_1 + E_1^{-1}}{4} , \quad \alpha_2^S = \frac{E_2 + E_2^{-1}}{4} , \quad \alpha_3^S = 0 .$$

For this case, we know that $\omega_p^* = 0$ from (2.8). It is easy to check that the 2nd level point SOR iteration becomes trivial and that only the 1st level block SOR iteration is necessary, which is identical to the traditional red/black SOR method with the following optimal relaxation parameter

$$\omega^* = \omega_b^* = \frac{2}{1 + [1 - \cos^2(\pi h)]^{1/2}} \approx 2 - 2\pi h . \qquad (4.1)$$

4.2 Rotated 5-point stencil

Another 5-point stencil discretization of the Poisson equation is [5]

$$\frac{1}{2h^2}\left(u_{j+1,k+1}+u_{j+1,k-1}+u_{j-1,k+1}+u_{j-1,k-1}-4u_{j,k}\right)=f_{j,k}\ ,$$

which is also a special case of (3.1) with

$$q_1=0\ ,\quad q_2=0\ ,\quad q_3=\frac{1}{2}\ ,\quad \text{and}\quad q=2\ .$$

Consequently, we find that

$$\alpha_1=0\ ,\quad \alpha_2=0\ ,\quad \alpha_3=\cos(\xi\pi h)\cos(\eta\pi h)\ ,$$

and

$$\alpha_1^S=0\ ,\quad \alpha_2^S=0\ ,\quad \alpha_3^S=\frac{(E_1+E_1^{-1})(E_2+E_2^{-1})}{4}\ .$$

It turns out that $\omega_b^*=1$, and in this case the 1st level block SOR iteration becomes trivial. Only the 2nd level point SOR iteration is necessary, which can be written as

$$u_{j,k}^{n+1}=(1-\omega^*)\,u_{j,k}^n+\omega^*(\alpha_3^S u_{j,k}^n-\frac{1}{2}h^2 f_{j,k})\qquad (j,k)\ \text{red}\ \textit{or}\ \text{black}$$

$$u_{j,k}^{n+1}=(1-\omega^*)\,u_{j,k}^n+\omega^*(\alpha_3^S u_{j,k}^{n+1}-\frac{1}{2}h^2 f_{j,k})\qquad (j,k)\ \text{orange}\ \textit{or}\ \text{green}$$

where

$$\omega^*=\omega_p^*=\frac{2}{1+[1-\cos^4(\pi h)]^{1/2}}\approx 2-2\sqrt{2}\pi h\ . \tag{4.2}$$

By comparing (4.1) and (4.2), we find that the only difference between the standard and rotated 5-point stencil discretizations is that the mesh size is h in the first case , and $\sqrt{2}h$ in the second case. The optimal relaxation parameter ω^* and spectral radius $\rho^*=\omega^*-1$ have therefore to be adjusted accordingly. Note however that the above observation depends on the isotropy of the Poisson equation, since the standard and rotated 5-point stencils give rise to different discretizations in the anisotropic case.

5. Convergence rate analysis

In this section, we show how to select the optimal relaxation parameters ω_b^* and ω_p^* for the 2-level 4-color SOR method described in Section 3, and we analyze the convergence rate of the resulting method when it is applied to equation (3.1) with nondegenerate coefficients, i.e., for

$$q_1 > 0, \quad q_2 > 0, \quad q_3 > 0.$$

5.1 Determination of optimal 2-level relaxation parameters

First, let us concentrate on the 2nd-level point iteration. In order to determine the optimal relaxation parameter, we need to find the spectral radius of the point Jacobi iteration which is given by

$$\mu_{p,max} = \max_{1 \le \xi,\eta \le N-1} |\alpha_3| = \frac{4q_3\cos^2(\pi h)}{q} \approx \frac{4q_3}{q}(1-\pi^2 h^2),$$

where the maximum value of $|\alpha_3|$ occurs for $(\xi,\eta) = (1,1)$ and $(N-1,N-1)$. Since the spectral radius of the point Jacobi iteration is bounded by the constant $\frac{4q_3}{q}$ which is less than 1, even a simple point Jacobi relaxation converges reasonably fast. Nevertheless, this can be further improved by a point SOR iteration using the following optimal relaxation parameter

$$\omega_p^* = \frac{2}{1+[1-\frac{(4q_3)^2\cos^4(\pi h)}{q^2}]^{1/2}} \approx 1 + \frac{1}{4}(\frac{4q_3}{q})^2, \tag{5.1}$$

with the spectral radius

$$\rho_p^* = \omega_p^* - 1 \approx \frac{1}{4}(\frac{4q_3}{q})^2. \tag{5.2}$$

For a typical example, we have $q_3 = 1$ and $q = 20$ (see Section 6) so that $\rho_p^* \approx 0.01$. Since the error can be damped approximately at the rate 10^{-2M}, where M is the

number of 2nd-level iterations, only 2 or 3 point SOR iterations inside each block SOR iteration are necessary. The fact that the 2nd-level point SOR iteration requires only a constant number M of steps to converge, where M is usually 2 or 3, plays a crucial role in our analysis of the convergence rate of the 2-level SOR method. By using this observation, it will be shown below that the convergence rate of the 2-level SOR scheme is similar to that of the standard SOR method for a 5-point stencil, or of the 9-point SOR scheme discussed in [2].

Next, we examine the 1st-level block iteration. The spectral radius of the block Jacobi iteration matrix (3.5) is given by

$$\mu_{b,max} = \max_{1\le\xi,\eta\le N-1} \{ | \frac{\alpha_1+\alpha_2}{1-\alpha_3} |, | \frac{\alpha_1-\alpha_2}{1+\alpha_3} | \} = \frac{2(q_1+q_2)\cos(\pi h)}{q-4q_3\cos^2(\pi h)} ,$$

which occurs at $(\xi,\eta) = (1,1), (1,N-1), (N-1,1)$ and $(N-1,N-1)$. By using the fact that $q = 2q_1+2q_2+4q_4$, we can simplify $\mu_{b,max}$ as

$$\mu_{b,max} = \frac{(q-4q_3)\cos(\pi h)}{q-4q_3\cos^2(\pi h)} \approx 1 - (\frac{1}{2}+\frac{4q_3}{q-4q_3})\pi^2 h^2 .$$

Hence, the optimal relaxation parameter for the block SOR iteration is

$$\omega_b^* = \frac{2}{1+(1-\mu_{b,max}^2)^{1/2}} \approx 2 - 2\,(1+\frac{8q_3}{q-4q_3})^{1/2}\pi h , \tag{5.3}$$

and the spectral radius is

$$\rho_b^* = \omega_b^* - 1 \approx 1 - 2\,(1+\frac{8q_3}{q-4q_3})^{1/2}\pi h . \tag{5.4}$$

Therefore, if $q_3 = 1$ and $q = 20$, then $\rho_b^* = 1 - \sqrt{6}\pi h$.

Since for a fixed point, the 2-level SOR method divides neighboring points into two groups and operates on one group at the block iteration level and on the other group at the point iteration level, and since each block SOR iteration at the first level

requires M point SOR iterations at the second level, it is convenient to define the *effective* number of iterations for one 2-level SOR iteration as

$$n_{eff} \equiv \frac{w_p M + w_b}{w_b + w_p} , \tag{5.5}$$

where w_b and w_p represent the amount of work required per block and per point iteration respectively. The number n_{eff} measures approximately the computational burden of one full 2-level SOR iteration in terms of equivalent 9-point Jacobi iterations.

If the point SOR iteration converges in M iterations, the convergence rate of the 2-level SOR method is then only determined by that of the block SOR iteration. Therefore, we can define the *effective* spectral radius of the 2-level SOR iteration as

$$\rho^*_{eff} \equiv (\rho^*_b)^{\frac{1}{n_{eff}}} = (\rho^*_b)^{\frac{w_b + w_p}{w_p M + w_b}} , \tag{5.6}$$

which is used to measure the average smoothing rate per effective iteration of the 2-level SOR scheme.

For the above example, since the amount of computational work for each block and point SOR iteration is the same, we have $w_p = w_b$, so that

$$n_{eff} = \frac{M+1}{2} , \quad \rho^*_{eff} \approx 1 - \frac{2}{M+1}\sqrt{6}\pi h .$$

When $M = 2$, we find therefore that

$$n_{eff} = \frac{3}{2} = 1.5 , \quad \rho^*_{eff} \approx 1 - 1.63\pi h . \tag{5.7}$$

The above effective spectral radius ρ^*_{eff} should be compared with the spectral radius $\rho^*_9 \approx 1 - 1.79\pi h$ that was obtained for the 9-point SOR method discussed by [2]. In the next section, we will present a 2-level SOR method with a different computational

ordering whose effective spectral radius is $\rho_{eff}^{*} \approx 1 - 2.26\pi h$.

We see from the above comparison that the 2-level SOR method and the 9-point SOR procedure of [2] have very similar convergence rates. The main difference is of course that the method of [2] is a single-level method which uses only one relaxation parameter ω^{*} . In addition, its convergence rate analysis requires the study of the solution of a quartic equation, and does not yield closed-form relations between ρ^{*} , ω^{*} and the spectral radius μ of the 9-point Jacobi iteration matrix. By comparison, the approach that we have used above to study the convergence of the 2-level SOR method relies on the standard SOR theory, and provides closed-form relations between ρ_{p}^{*}, ω_{p}^{*}, and $\mu_{p,max}$, and between ρ_{b}^{*}, ω_{b}^{*}, and $\mu_{b,max}$.

Finally, note that the amount of work required by each effective iteration for the 9-point stencil case is about twice as large as for a standard 5-point SOR iteration. Thus, to compare the convergence rate of the 2-level SOR method with that of the standard 5-point SOR scheme, we must compare ρ_{eff}^{*} with the spectral radius $(\rho_{5}^{*})^{2} \approx 1 - 4\pi h$ corresponding to two 5-point SOR iterations. This comparison seems to indicate that the 5-point SOR iteration converges faster than the 2-level SOR method, or the 9-point SOR method discussed in [2]. However, the 9-point stencil discretization is more accurate than the corresponding 5-point stencil discretization. Thus, for the same accuracy, we can select h larger for the 9-point stencil discretization so that in actuality the 2-level or single-level 9-point SOR methods may converge faster than the standard 5-point SOR method.

5.2 Computational order

In the above discussion, we have used a particular computational order, i.e., { red → orange → black → green }. Now, let us consider other computational orderings. Although there exist $4! = 24$ different ways to permute the computational order for these 4 colors, they only result in 3 different 2-level SOR iteration schemes. By interchanging the relative positions of α_1, α_2, and α_3 in the matrix $A(\xi,\eta)$, we can obtain only 6 different matrices, each of which corresponds to 4 different computational orderings. Furthermore, we can divide these 6 matrices into 3 classes:

Class 1 :
$$\begin{bmatrix} 1 & -\alpha_1 & -\alpha_2 & -\alpha_3 \\ -\alpha_1 & 1 & -\alpha_3 & -\alpha_2 \\ -\alpha_2 & -\alpha_3 & 1 & -\alpha_1 \\ -\alpha_3 & -\alpha_2 & -\alpha_1 & 1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 1 & -\alpha_1 & -\alpha_3 & -\alpha_2 \\ -\alpha_1 & 1 & -\alpha_2 & -\alpha_3 \\ -\alpha_3 & -\alpha_2 & 1 & -\alpha_1 \\ -\alpha_2 & -\alpha_3 & -\alpha_1 & 1 \end{bmatrix}$$

Class 2 :
$$\begin{bmatrix} 1 & -\alpha_2 & -\alpha_1 & -\alpha_3 \\ -\alpha_2 & 1 & -\alpha_3 & -\alpha_1 \\ -\alpha_1 & -\alpha_3 & 1 & -\alpha_2 \\ -\alpha_3 & -\alpha_1 & -\alpha_2 & 1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 1 & -\alpha_2 & -\alpha_3 & -\alpha_1 \\ -\alpha_2 & 1 & -\alpha_1 & -\alpha_3 \\ -\alpha_3 & -\alpha_1 & 1 & -\alpha_2 \\ -\alpha_1 & -\alpha_3 & -\alpha_2 & 1 \end{bmatrix}$$

Class 3 :
$$\begin{bmatrix} 1 & -\alpha_3 & -\alpha_1 & -\alpha_2 \\ -\alpha_3 & 1 & -\alpha_2 & -\alpha_1 \\ -\alpha_1 & -\alpha_2 & 1 & -\alpha_3 \\ -\alpha_2 & -\alpha_1 & -\alpha_3 & 1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 1 & -\alpha_3 & -\alpha_2 & -\alpha_1 \\ -\alpha_3 & 1 & -\alpha_1 & -\alpha_2 \\ -\alpha_2 & -\alpha_1 & 1 & -\alpha_3 \\ -\alpha_1 & -\alpha_2 & -\alpha_3 & 1 \end{bmatrix}$$

It is easy to see that the same 2-level SOR method applies to matrices within the same class. Although the discussion in section 5.1 applies only to matrices of class 3, we can use a similar approach to obtain optimal block and point relaxation parameters and spectral radii for a 2-level SOR method for matrices of Classes 1 and 2. For matrices of Class 1, we find

$$\omega_p^* \approx 1 + \frac{1}{4}(\frac{2q_1}{q})^2 , \quad \rho_p^* \approx \frac{1}{4}(\frac{2q_1}{q})^2 , \tag{5.8}$$

$$\omega_b^* \approx 2 - 2\,(\frac{q+4q_3}{q-2q_1})^{1/2}\pi h , \quad \rho_b^* \approx 1 - 2\,(\frac{q+4q_3}{q-2q_1})^{1/2}\pi h . \tag{5.9}$$

and for matrices of Class 2, we need only to replace q_1 by q_2 in the above expressions.

The data flow diagram for the computational order { red → black → green → orange }, which corresponds to a 2-level SOR method applied to matrices of Class 1, is shown in Figure 3. Let us analyze the convergence rate for this 2-level SOR iteration. From Figure 3, it is easy to see that $w_p = \frac{1}{3} w_b$. Therefore, from (5.5) and (5.6), we have

$$n_{eff} = \frac{3+M}{4} , \quad \rho_{eff}^* = (\rho_b^*)^{\frac{4}{3+M}} .$$

Consider now the typical example where $q_1 = q_2 = 4$ and $q_3 = 1$. By using (5.8) and (5.9), we find that the spectral radius of the point SOR iteration becomes larger, but the spectral radius of the block SOR iteration becomes smaller, i.e.

$$\rho_p^* \approx 4 \times 10^{-2} , \quad \rho_b^* \approx 1 - \sqrt{8}\pi h .$$

Therefore, the effective spectral radius can be expressed as

$$\rho_{eff}^* \approx 1 - \frac{4}{M+3}\sqrt{8}\pi h .$$

This gives

$$\rho_{eff}^* \approx 1 - 2.26\pi h \quad \text{if } M = 2, \quad \rho_{eff}^* \approx 1 - 1.89\pi h \quad \text{if } M = 3. \tag{5.10}$$

By comparing (5.7) and (5.10), we observe that the performance of a 2-level SOR iteration applied to matrices of the first or second class is in fact better for this specific example.

6. Numerical Examples

We consider the system of equations obtained from a 9-point stencil discretization of the isotropic Poisson equation, i.e.,

$$\frac{1}{6h^2}\{4(u_{j+1,k}+u_{j-1,k})+4(u_{j,k+1}+u_{j,k-1})+(u_{j+1,k+1}+u_{j+1,k-1}$$

$$+u_{j-1,k+1}+u_{j-1,k-1})-20\,u_{j,k}\}=f_{j,k} \quad j,k=1,2,\cdots,N-1, \qquad (6.1)$$

with zero boundary conditions and $h=\frac{1}{N}=\frac{1}{20}$. In this case, $q_1=q_2=4$, and $q_3=1$. Since in this example the performance of the 2-level SOR method for matrices $A(\xi,\eta)$ of Classes 1 and 2 is the same, we compare only the following two computational orders:

order (a): { red → orange → black → green },

order (b): { red → black → green → orange }.

The computational orders (a) and (b) are obtained by applying the 2-level SOR iteration to matrices $A(\xi,\eta)$ belonging respectively to Classes 3 and 1. Their spectral radii and optimal relaxation parameters for the block SOR and the point SOR iterations are summarized in Table 2.

order	ω_b^*	ρ_b^*	ω_p^*	ρ_p^*
(a)	1.679931	0.679931	1.009702	0.009702
(b)	1.640105	0.640105	1.042400	0.042400

Table 2

We use the following two test problems:

Example 1: The driving function is $e^{5x}\,[2x(x-1)+y(y-1)(25x^2-5x-8)]$ and the true solution is $e^{5x}\,x(x-1)y(y-1)$. In this case, the solution is a smooth function

with a wideband 2-D Fourier spectrum which is concentrated in the region where ξ and η are small.

Example 2: The driving function is $-74\pi^2\sin(5\pi x)\sin(7\pi y)$ and the true solution is $\sin(5\pi x)\sin(7\pi y)$. This corresponds to the case when the solution is a rapidly oscillatory function containing a single Fourier component at $(\xi,\eta) = (5,7)$.

The computed results are shown in Figures 5 and 6, where we plot the maximum error at each iteration as a function of the number of block SOR iterations. Each curve is parameterized by the number M of point SOR iterations that we have used. It is almost impossible to distinguish the curves with $M = 2,3,4$ for computational order (a) in both examples. Hence, it is reasonable to choose $M = 2$ in this case. When the computational order (b) is applied to the first example, where the solution contains low frequency components, the curve for $M = 3$ is slightly better than for $M = 2$. Nevertheless, the difference is very small. For the second example, the curves with $M = 2,3,4$ are in fact not distinguishable. Thus, for computational order (b), it is still preferable to choose $M = 2$, since less computations are required.

To demonstrate the convergence rate of the 2-level SOR method, we choose another test problem with zero driving function and boundary conditions. This is in fact a homogeneous Laplace equation and its solution is zero. Two initial guesses are considered: (1) a smooth function which is chosen to be $x(x-1)y(y-1)$ and (2) a random 2D sequence. In Figure 7, we plot the 2-norm of the error versus the effective number (n_{eff}) of iterations for the above two computational orders and $M=2$. The results show that the 2-level SOR method with computational order (b) is better than that with order (a) and that the convergence rate of 2-level SOR method is not

sensitive to the smoothness of the initial errors. Since the problem with initial guess $x(x-1)y(y-1)$ was also used to demonstrate the convergence rate of Adams et al.'s SOR method in [2], we are able to compare the convergence rates of our method with theirs for this test problem. It turns out that these two methods have very similar convergence rates.

7. Conclusions and Generalizations

In this paper, we have transformed the system of equations for a discretized elliptic PDE from the space domain to the frequency domain so that we were able to interpret the SOR method from a new point of view. This new formulation has helped us to design a 2-level SOR method with optimal block and point relaxation parameters. The resulting 2-level 4-color SOR method for the 9-point stencil discretization of the Poisson equation was shown to be efficient with spectral radius $1 - C\pi h$, and numerical examples confirm our analysis.

The constant C of Adams et al.'s SOR method with various orderings and the line SOR method was compared in [2]. The results for the 9-point stencil discretization can be summarized as follows. The constant C ranges from 1.6 to 2.45 for Adams et al.'s method, $C = 1.63$ or 2.26 for the 2-level method, and $C = 2.82$ for the line SOR method. In practice, when the initial error is smooth, the convergence rate of the 2-level SOR method is similar to that of Adams et al.'s method. By comparing the constant C, we see that the line SOR method is slightly faster than both the 2-level and Adams et al.'s methods. However, it should be emphasized that the line SOR method is less parallelizable since it needs a sequential direct method to solve tridiagonal matrix equations which describe the coupling between points of each line. Thus, from a parallel processing point of view, the 2-level and Adams et al.'s SOR methods are more attractive.

The 2-level SOR iteration method presented here can be generalized easily to higher-dimensional problems. A 3-level 8-color SOR scheme can be described as follows. Consider a nondegenerate 27-point discretization of the 3-D Poisson equation.

Suppose that each grid point is indexed by (j,k,l). We can label these points with 8 colors depending on whether j, k, and l are even or odd. Following a procedure similar to the one used in Section 3, we transform the discretized system from the space domain to the frequency domain so that in the frequency domain we obtain a discretization matrix which is block diagonal with 8×8 block matrices along the diagonal. Each of these blocks describes the coupling of the Fourier components of the 8 colors at a fixed frequency. Since the discretization scheme is nondegenerate, each 8×8 matrix block is full. In order to apply the SOR method for each of these 8×8 matrices, we can block partition them into 4×4 submatrices. This results in a 1st level block SOR iteration. However, the 1st level block SOR iteration requires inverting 4×4 full matrices, which can be accomplished by performing several 2nd level block SOR and 3rd level point SOR iterations. Note that both the 2nd level block SOR and 3rd level point SOR iterations require a constant number of steps to converge. The total number of iterations required by the above 3-level SOR method, which is $O(\frac{1}{h})$, is therefore determined primarily by the convergence rate of the 1st level block SOR iteration.

There are many different possible computational orders for the above 3-level SOR procedure. A typical one can be chosen as follows. At the 1st level, we can distinguish two big blocks depending on whether $(j+k+l)$ is even or odd. At the 2nd level, within each big block, points are further divided into two smaller blocks according to whether $(j+k)$ is even or odd. Finally, at the 3rd level, each color can be separated from each other.

It is straightforward to generalize the above procedure to obtain an n-D n-level 2^n-color SOR method. Here, we have considered the case where $n = 2$.

Another generalization of interest would be to extend the 2-level SOR iteration procedure described in this paper to PDEs with space-varying coefficients. It is natural in this context to combine the 2-level SOR method discussed here with the local relaxation procedure developed in [4], [6] and [9]. The main idea of the local relaxation method can be roughly stated as follows. Each local finite difference equation is viewed as if it were homogeneous over the entire problem domain so that at each point a local relaxation parameter is determined on the basis of the local coefficients of the PDE and of the boundary conditions for the whole domain. Hence, a 2-level local relaxation method would use the local coefficients and boundary conditions to choose optimal local block and point relaxation parameters at each grid point, so that different grid points would have therefore different block and point relaxation parameters.

Note that the pure frequency domain approach described in this paper depends heavily on the specific coloring and partitioning scheme that we have used. The relation existing between the single-level rowwise and multicolor SOR methods for the 5-point stencil and the 9-point stencil cases can be explained by introducing a tilted grid [10][2]. There does not seem to be an easy way to apply the tilted grid concept to obtain a 2-level rowwise SOR method.

Acknowledgement

The authors would like to thank Professor Lloyd N. Trefethen for his comments at various stages of this work. The authors are also grateful to the referees for several valuable suggestions, and in particular for pointing out the rederivation of Adams et al.'s SOR method presented in Section 3.4.

References

[1] L. Adams and H. F. Jordan, "Is SOR Color-Blind," *SIAM J. Sci. Stat.*, vol. 7, no. 2, Apr. 1986.

[2] L. Adams, R. J. LeVeque, and D. M. Young, "Analysis of the SOR Iteration for the 9-Point Laplacian," To appear in *SIAM J. Num. Analy.*.

[3] L. Adams and J. M. Ortega, "A Multi-Color SOR Method for Parallel Computation," ICASE report, 82-9, Apr. 1982.

[4] E. F. Botta and A. E. P. Veldman, "On Local Relaxation Methods and Their Application to Convection-Diffusion Equations," *Journal of Computational Physics*, vol. 48, pp. 127-149, 1981.

[5] G. Dahlquist, A. Bjorck, and N. Anderson, *Numerical Methods*. Englewood Cliffs, N.J. : Prentice-Hall, Inc. , 1974.

[6] L. W. Erhlich, "The Ad-hoc SOR Method: a Local Relaxation Scheme," in *Elliptic Problem Solvers II*, Academic Press, 1984, pp. 257-269.

[7] S. P. Frankel, "Convergence Rates of Iterative Treatments of Partial Differential Equations," *Math. Tables Aids Comput.*, vol. 4, pp. 65-75, 1950.

[8] P. R. Garabedian, "Estimation of the Relaxation Factor for Small Mesh Size," *Math. Tables Aids Comput.*, vol. 10, pp. 183-185, 1956.

[9] C.-C. J. Kuo, B. C. Levy, and B. R. Musicus, "A Local Relaxation Method for Solving Elliptic PDEs on Mesh-Connected Arrays," *SIAM J. Sci. Stat. Comp.*, vol. 8, no. 4, pp. 530-573, Jul. 1987.

[10] R. J. LeVeque and L. N. Trefethen, "Fourier Analysis of the SOR Iteration," Numerical Analysis Report 86-6, Department of Mathematics, MIT, Cambridge, MA. and also ICASE Report No. 86-93., Sep. 1986.

[11] R. S. Varga, *Matrix Iterative Analysis*. Englewood Cliffs, N.J. : Prentice-Hall, Inc. , 1962.

[12] D. M. Young , "Iterative Methods for Solving Partial Differential Equations of Elliptic Type ," Doctoral Thesis , Harvard University , 1950.

[13] D. M. Young, *Iterative Solution of Large Linear Systems*. New York, N.Y.: Academic Press, Inc., 1971.

Figure Captions

Figure 1: 4-color partitioning for the 9-point stencil discretization.

Figure 2: Data flow diagram for a 2-level 4-color SOR method with computational order { red → orange → black → green }. Step 1: first half of a block SOR iteration. Step 2(a) and (b): one point SOR iteration for red and orange points. Step 3: second half of a block SOR iteration. Step 4(a) and (b): one point SOR iteration for black and green points.

Figure 3: Data flow diagram for a 2-level 4-color SOR method with computational order { red → black → green → orange }. Step 1: first half of a block SOR iteration. Step 2(a) and (b): one point SOR iteration for red and black points. Step 3: second half of a block SOR iteration. Step 4(a) and (b): one point SOR iteration for orange and green points.

Figure 4: Another 4-color partitioning scheme.

Figure 5: Computer simulation results for Example 1 with computational orders (a) { red → orange → black → green } and (b) { red → black → green → orange }. The x-axis is the number of 1st-level block iterations and the y-axis is the maximum error at each iteration.

Figure 6: Computer simulation results for Example 2 with computational orders (a) { red → orange → black → green } and (b) { red → black → green → orange }. The x-axis is the number of 1st-level block iterations and the y-axis is the maximum error at each iteration.

Figure 7: Convergence history (2-norm of the error versus the number of effective iterations) for computational orders (a) { red → orange → black → green } and (b) { red → black → green → orange } with $M=2$. The driving function is zero and the initial values are (1) $x(x-1)y(y-1)$ and (2) a random sequence.

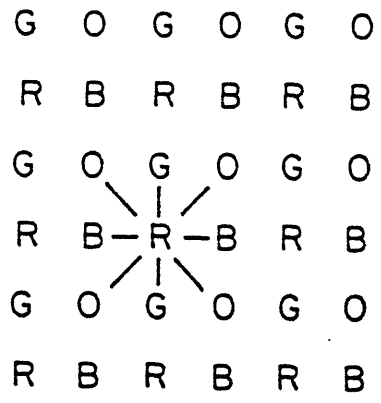

Figure 1: 4-color partitioning for the 9-point stencil discretization.

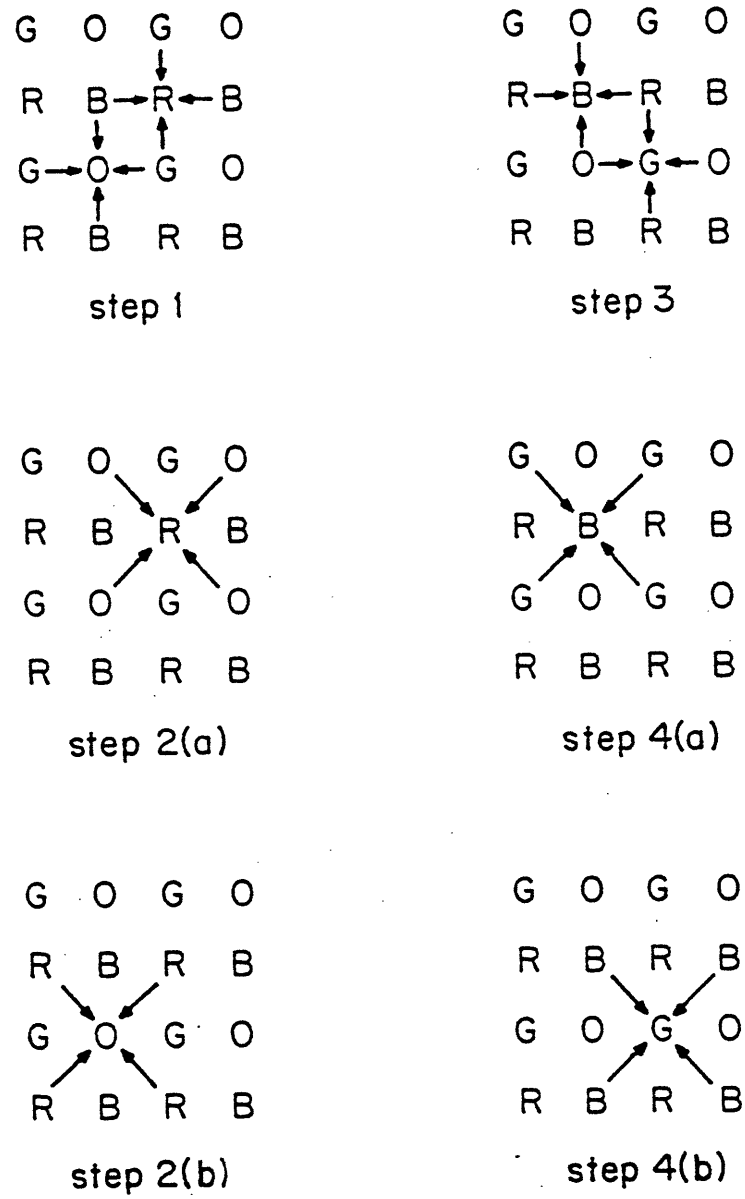

Figure 2: Data flow diagram for a 2-level 4-color SOR method with computational order { red → orange → black → green }. Step 1: first half of a block SOR iteration. Step 2(a) and (b): one point SOR iteration for red and orange points. Step 3: second half of a block SOR iteration. Step 4(a) and (b): one point SOR iteration for black and green points.

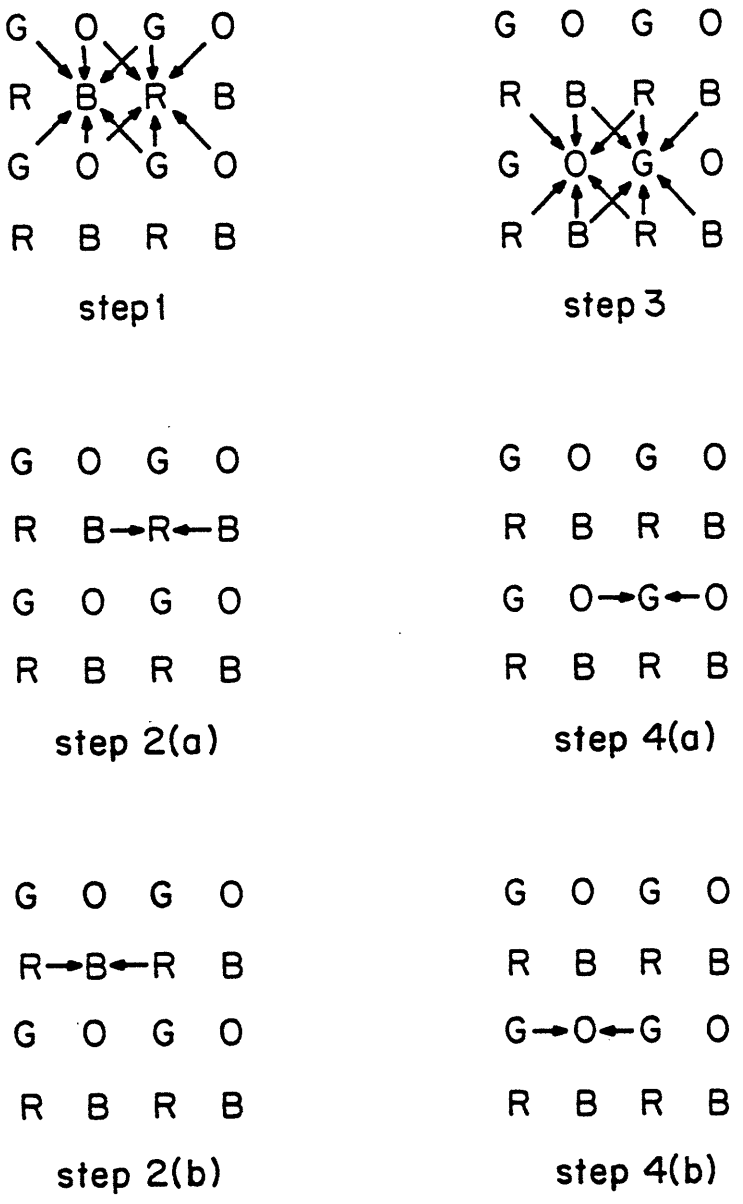

Figure 3: Data flow diagram for a 2-level 4-color SOR method with computational order { red → black → green → orange }. Step 1: first half of a block SOR iteration. Step 2(a) and (b): one point SOR iteration for red and black points. Step 3: second half of a block SOR iteration. Step 4(a) and (b): one point SOR iteration for orange and green points.

G O R B G O
R B G O R B
G O R B G O
R B G O R B
G O R B G O
R B G O R B

Figure 4: Another 4-color partitioning scheme.

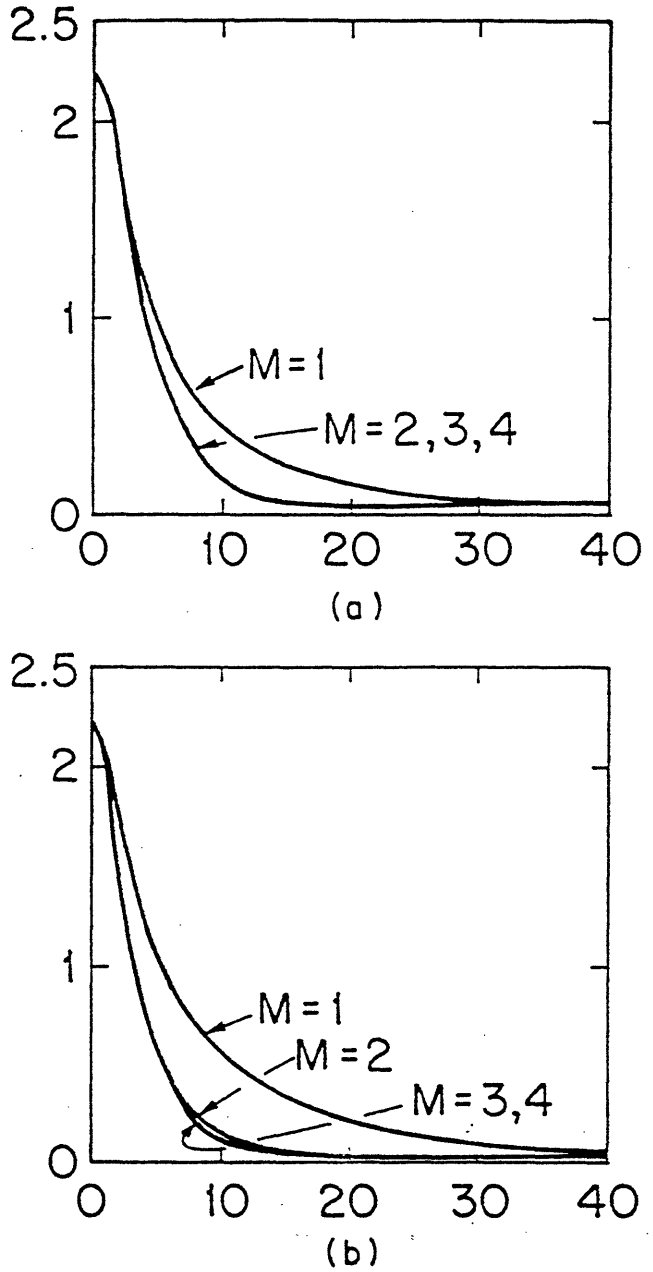

Figure 5: Computer simulation results for Example 1 with computational orders (a) { red → orange → black → green } and (b) { red → black → green → orange }. The x-axis is the number of 1st-level block iterations and the y-axis is the maximum error at each iteration.

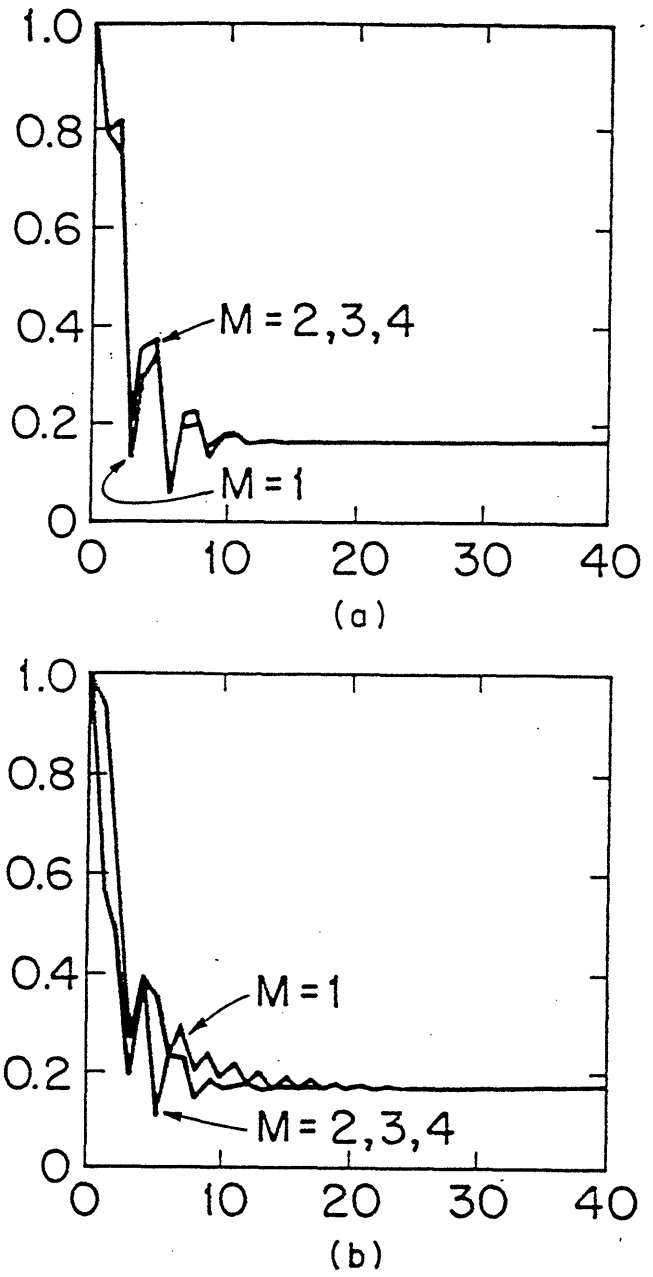

Figure 6: Computer simulation results for Example 2 with computational orders (a) { red → orange → black → green } and (b) { red → black → green → orange }. The x-axis is the number of 1st-level block iterations and the y-axis is the maximum error at each iteration.

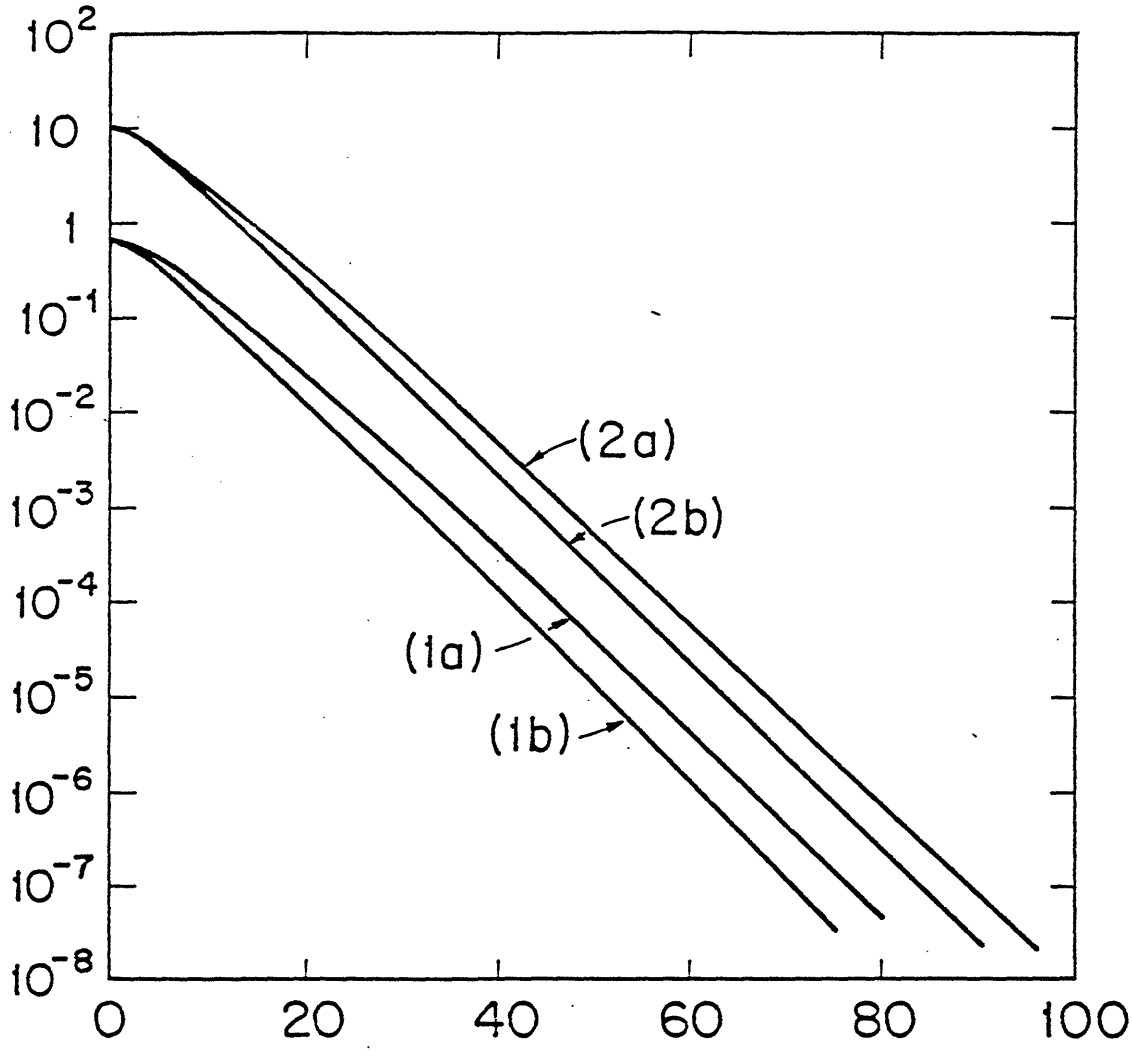

Figure 7: Convergence history (2-norm of the error versus the number of effective iterations) for computational orders (a) { red → orange → black → green } and (b) { red → black → green → orange } with $M=2$. The driving function is zero and the initial values are (1) $x(x-1)y(y-1)$ and (2) a random sequence.