SIAM J. SCI. STAT. COMPUT.
Vol. 8, No. 4, July 1987

A LOCAL RELAXATION METHOD FOR SOLVING ELLIPTIC PDEs ON MESH-CONNECTED ARRAYS*

C.-C. JAY KUO†, BERNARD C. LEVY† AND BRUCE R. MUSICUS‡

Abstract. A local relaxation method for solving linear elliptic PDEs with $O(N)$ processors and $O(\sqrt{N})$ computation time is proposed. We first examine the implementation of traditional relaxation algorithms for solving elliptic PDEs on mesh-connected processor arrays, which require $O(N)$ processors and $O(N)$ computation time. The disadvantage of these implementations is that the determination of the acceleration factors requires some global communication at each iteration. The high communication cost increases the computation time per iteration significantly. Therefore, a local relaxation scheme is proposed to achieve the acceleration effect with very little global communication in the loading stage. We use a Fourier analysis approach to analyze the local relaxation method and also show its convergence. The convergence rate of the local relaxation method is studied by computer simulation.

Key words. mesh-connected processor arrays, elliptic partial differential equations, successive over-relaxation, local relaxation, Fourier analysis, parallel computation

AMS(MOS) subject classifications. 65N20, 65F10

1. Introduction.

Consider a 2-D linear elliptic PDE on a unit square discretized by a finite-difference method with a uniform grid. There is a finite difference equation associated with each grid point, so that a system of linear equations is obtained by this discretization procedure. We may assign one processor to each grid point and connect every processor to its four nearest neighbors. This kind of computer architecture, known as a mesh-connected processor array, suggests a natural parallel computation scheme to solve the above system of equations, i.e., parallel computation in the space domain. Jacobi and Gauss–Seidel relaxation methods seem particularly suitable for mesh-connected processors, since each processor uses only the most recent values computed by its neighbors to update its own value. Unfortunately, the convergence rate of these algorithms is slow. The convergence rate can be improved by various acceleration schemes such as successive over-relaxation (SOR) and Chebyshev semi-iterative relaxation (CSI) [17]. However, to obtain the acceleration effect requires that the acceleration factors should be estimated adaptively [8]. This procedure requires global communication on a mesh-connected processor array and increases the computation time per iteration enormously. Any time savings due to acceleration may be cancelled out by the increased communication time. In order to improve the convergence rate as well as to avoid global communication, a recently developed approach known as the ad hoc SOR [5], [6], or local relaxation [3] method seems to be useful.

The local relaxation scheme was found empirically by Ehrlich [5], [6] and Botta and Veldman [3]. They applied this method to a very broad class of problems and found its efficiency by studying many numerical examples. In this paper, we approach the same problem from an analytical point of view, clearly prove the convergence of

* Received by the editors October 21, 1985; accepted for publication (in revised form) April 18, 1986. This work was supported in part by National Aeronautics and Space Administration grant NAGW-448, Army Research Office grant DAAG29-84-K-0005, the Advanced Research Projects Agency monitored by the Office of Naval Research under contract N00014-81-K-0742, and Air Force Office of Scientific Research contract F49620-84-C-0004.

† Laboratory for Information and Decision Systems and Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139.

‡ Research Laboratory of Electronics and Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139.

this method for the case of symmetric positive-definite matrices and provide an analytical explanation for the good performance of the method.

The conventional way to analyze the SOR method is to use matrix analysis [17]. This approach depends heavily on the ordering of the grid points and on the properties of the resulting sparse matrix. An alternative technique, which was employed in [12], [14] and [15] to analyze relaxation algorithms, is to use Fourier analysis. Strictly speaking, Fourier analysis applies only to linear constant coefficient PDEs on an infinite domain, or with periodic boundary conditions. Nevertheless, at a heuristic level this approach provides a useful tool for the analysis of more general PDE problems, and it has been used by Brandt [4] to study the error smoothing effect of relaxation algorithms and to develop multigrid methods. Since then, the Fourier analysis approach has received a large amount of attention in the study of multigrid methods [16]. Following the same idea, we shall apply the Fourier analysis approach to the SOR method. For the Poisson Problem defined on the unit square with Dirichlet boundary conditions, we obtain the same result as Young's SOR method. However, our derivation is simpler. For space-varying PDEs, the local relaxation scheme uses space adaptive relaxation parameters. This is different from Young's SOR method which uses time adaptive relaxation parameters [8].

The paper is organized as follows. Section 2 discusses the implementation of the Jacobi, Gauss–Seidel, adaptive SOR and local relaxation methods on mesh-connected processor arrays. Section 3 proves the convergence of the local relaxation method. The Fourier analysis approach is used to analyze the local relaxation method for 5-point and 9-point stencils, respectively, in §§ 4 and 5. It turns out that the 9-point stencil analysis requires a slight modification of the basic convergence result of § 3. Section 6 shows the results of a computer simulation on a test problem which indicates that the convergence rate of the local relaxation method is superior to that of the adaptive SOR method. Some further extensions and conclusions are mentioned in § 7.

2. Implementing numerical PDE algorithms on mesh-connected processor arrays. The computation time for an iterative algorithm equals the product of the number of iterations and time per iteration. In a sequential machine, time per iteration is determined by operation counts, especially by the number of floating point operations required. In a multiprocessor machine, however, time per iteration depends heavily on the algorithm chosen and on the communication scheme. Consider mesh-connected processor arrays in which processors are organized in a geometrically regular two-dimensional square "tiling" pattern and connected only to their nearest neighbors. Algorithms using only local communication will take $O(1)$ communication time, while those using global communication require $O(\sqrt{N})$ communication time per iteration since for an array with N processors, communications between processors located on opposite sides of the array will take $O(\sqrt{N})$ time. We thus seek algorithms with fast convergence rate, short computation time, and primarily local communication.

Let us use an example to illustrate how the above considerations affect the implementation of various iterative algorithms. Consider a self-adjoint second-order linear PDE defined on a closed unit square $\Omega = [0, 1] \times [0, 1]$,

$$-\frac{\partial}{\partial x_1}\left\{p(x_1, x_2)\frac{\partial u}{\partial x_1}\right\} - \frac{\partial}{\partial x_2}\left\{q(x_1, x_2)\frac{\partial u}{\partial x_2}\right\} + \sigma(x_1, x_2)u = f(x_1, x_2), \tag{2.1a}$$

$$(x_1, x_2) \in \Omega,$$

with the following boundary condition on Γ, the boundary of Ω,

$$\alpha(x_1, x_2)u + \beta(x_1, x_2)\frac{\partial u}{\partial n} = \gamma(x_1, x_2), \qquad (x_1, x_2) \in \Gamma, \tag{2.1b}$$

where $\partial u/\partial n$ denotes the outward derivative normal to Γ. The coefficient functions are assumed to be smooth and to satisfy

$$p(x_1, x_2) > 0, \quad q(x_1, x_2) > 0, \quad \sigma(x_1, x_2) \geqq 0, \quad (x_1, x_2) \in \Omega,$$

$$\alpha(x_1, x_2) \geqq 0, \quad \beta(x_1, x_2) \geqq 0, \quad \alpha + \beta > 0, \quad (x_1, x_2) \in \Gamma.$$

If we discretize (2.1) on a $\sqrt{N} \times \sqrt{N}$ uniform grid [17], we get a 5-point stencil; and the finite difference equation at an interior point (i, j) can be written as

$$d_{i,j}u_{i,j} - r_{i,j}u_{i+1,j} - l_{i,j}u_{i-1,j} - t_{i,j}u_{i,j+1} - b_{i,j}u_{i,j-1} = s_{i,j}, \tag{2.2}$$

with

$$l_{i,j} = p_{i-1/2,j}, \quad r_{i,j} = p_{i+1/2,j}, \quad b_{i,j} = q_{i,j-1/2}, \quad t_{i,j} = q_{i,j+1/2}, \tag{2.3a}$$

$$d_{i,j} = p_{i-1/2,j} + p_{i+1/2,j} + q_{i,j-1/2} + q_{i,j+1/2} + \sigma_{i,j}h^2, \qquad s_{i,j} = f_{i,j}h^2 \tag{2.3b}$$

where h is the grid spacing and $p_{i,j}$ is defined as $p(ih, jh)$. Similar discretized equations can be obtained for the boundary points where $u_{i,j}$ is unknown. Let us choose a particular order for those equations, and construct vectors u and s from the variables $u_{i,j}$ and $s_{i,j}$ arranged in the selected order; then the interior and boundary equations can be arranged in matrix form

$$Au = s, \tag{2.4}$$

where A contains the coefficients $d_{i,j}$, $l_{i,j}$, $r_{i,j}$, $t_{i,j}$ and $b_{i,j}$. The matrix A is symmetric, since $l_{i+1,j} = r_{i,j}$ and $b_{i,j+1} = t_{i,j}$. In addition, A is positive definite, since it is irreducibly diagonal dominant [17, p. 23].

Starting from equation (2.2), we can discuss the details of implementing different iterative algorithms with a mesh-connected processor array. Assume that we build a $\sqrt{N} \times \sqrt{N}$ grid of processors, and assign the processor at coordinate (i, j) the responsibility of calculating the value of $u_{i,j}$. Direct communication is allowed only between neighboring processors. At iteration $n+1$, each processor may combine the estimated value of $u^{(n)}$ in neighboring processors, together with its own estimate of $u_{i,j}^{(n)}$, in order to develop a new estimate $u_{i,j}^{(n+1)}$. A particularly simple iteration, for example, is the Jacobi method, in which we iteratively calculate:

$$u_{i,j}^{(n+1)} = d_{i,j}^{-1}(l_{i,j}u_{i-1,j}^{(n)} + r_{i,j}u_{i+1,j}^{(n)} + b_{i,j}u_{i,j-1}^{(n)} + t_{i,j}u_{i,j+1}^{(n)} + s_{i,j}).$$

According to the above iterative equation, each processor uses the values of $u^{(n)}$ obtained by its nearest neighbors to update its value at the current iteration. Time per iteration is constant, because both the communication time and computation time are constant.

If the grid point (i, j) is called a red point when $i+j$ is even, and a black point when $i+j$ is odd, the Jacobi method can be viewed in space and time as consisting of two interleaved, and totally independent, *computational waves* alternating between red and black points. This phenomenon is illustrated in Fig. 1, where the one-dimensional grid with red/black partitioning is shown in the horizontal direction while the evolution from one iteration to the next is indicated in the vertical direction. The solid and dotted lines represent two value-updating processes evolving with time, or two computational waves. In fact, these two waves result in unnecessary redundancy. We need only one wave to get the answer, since both waves converge to the same final values. If we delete one computational wave, the rate of utilization of the processors becomes one half, i.e., every processor works only half of the time. Therefore, we may

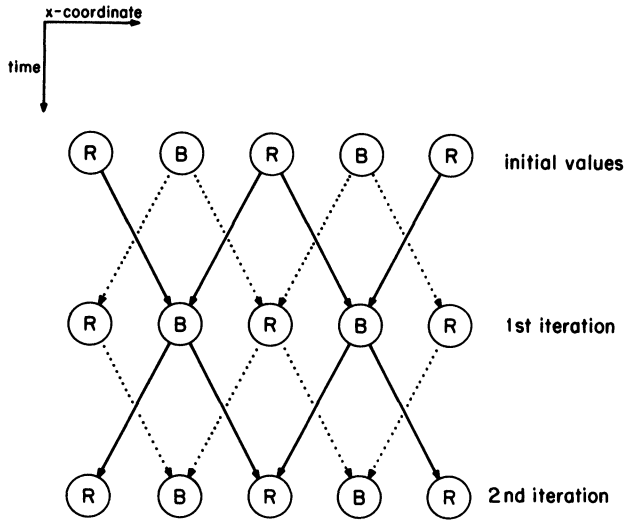


FIG. 1. 1-D *Jacobi relaxation with red/black partitioning.*

group one red and one black point together and assign them to a single processor. This saves half of the hardware cost without loss of computational efficiency (see Fig. 2).

To derive a Gauss-Seidel algorithm for this problem, let us consider the same red/black point partitioning and write the local equation as follows:

red points $(i+j$ *is even*):

$$u_{i,j}^{(n+1)} = d_{i,j}^{-1}(l_{i,j}u_{i-1,j}^{(n)} + r_{i,j}u_{i+1,j}^{(n)} + b_{i,j}u_{i,j-1}^{(n)} + t_{i,j}u_{i,j+1}^{(n)} + s_{i,j});$$

black points $(i+j$ *is odd*):

$$u_{i,j}^{(n+1)} = d_{i,j}^{-1}(l_{i,j}u_{i-1,j}^{(n+1)} + r_{i,j}u_{i+1,j}^{(n+1)} + b_{i,j}u_{i,j-1}^{(n+1)} + t_{i,j}u_{i,j+1}^{(n+1)} + s_{i,j}).$$

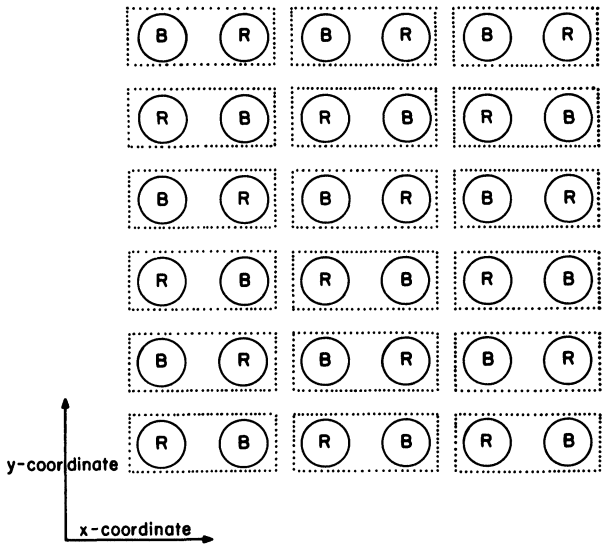


FIG. 2. 2-D *red/black partitioning and grouping.*

Other partitionings will lead to different Gauss-Seidel Schemes; however, the red/black partitioning approach is preferred for parallel implementation on mesh-connected arrays, because of its efficiency and simplicity. Note that the difference between the Jacobi and the Gauss-Seidel relaxation methods is that the Jacobi method updates the values of all nodes at one iteration while the Gauss-Seidel method updates the values of half of these nodes during a first step and updates the values of the other half during a second step based on the previously updated information, and these two steps form a complete iteration. For the case of a one-dimensional grid, we find that the Gauss-Seidel iteration is equivalent to the computational wave of the Jacobi iteration shown by the dotted line in Fig. 1. Therefore, we can save one half of the computational work by using Gauss-Seidel iteration on either a single processor or a mesh-connected array.

The chief shortcoming of the Jacobi or Gauss-Seidel iterative methods lies in their slow convergence rate. It usually happens that the spectral radius of the relaxation matrix is very close to 1, which causes the convergence rate to be extremely slow. The number of iterations needed is proportional to $O(N)$ [17]. Since time per iteration is constant, the total running time is also proportional to $O(N)$.

By applying different acceleration schemes to the Jacobi and Gauss-Seidel techniques, we can derive a variety of accelerated relaxation algorithms. Two typical examples are the Chebyshev semi-iterative (CSI) method and the successive over-relaxation (SOR) method. These acceleration schemes use carefully chosen relaxation parameters to reduce the spectral radii of the iterative matrices so that the iterative algorithms converge faster. To determine the relaxation parameters, CSI acceleration uses knowledge of the largest and smallest eigenvalues of the basic relaxation matrix and SOR acceleration uses knowledge of the spectral radius of the basic relaxation matrix [17]. For a given mesh-connected processor array, if we know these quantities a priori and broadcast them to all processors in the loading stage, each processor can compute the acceleration parameters on its own without additional communication cost. In this case, although the accelerated schemes require a little more computation and memory than the basic Jacobi and Gauss-Seidel relaxation schemes, they present some significant advantages. The reason is that the number of iterations needed is reduced tremendously, becoming $O(\sqrt{N})$ for both acceleration schemes [8]. However, in general we do not know the eigenvalues of the basic relaxation matrix in advance and have to estimate them by some adaptive procedure. To our knowledge, all the estimation procedures developed require the computation of the norms of some global vectors. Therefore, global communication cannot be avoided. This means that the communication cost for a single iteration in a mesh-connected array becomes $O(\sqrt{N})$. As a consequence, time per iteration is $O(\sqrt{N})$ and the total running time becomes $O(N)$ again.

Comparing this result with the result obtained for the basic relaxation methods, it seems that we do not benefit from acceleration schemes when we seek to implement iterative algorithms in parallel on mesh-connected arrays. This can be easily explained by noting that for a single processor, there is no distinction between local and global communications, since all data are fetched from the same memory, while for a mesh-connected processor array, long range communication costs much more than short range communication.

In addition to the above relaxation algorithms, another important class of algorithms for solving systems of linear equations can be derived from an optimization principle. The conjugate gradient (CG) algorithm is an example [8]. Without considering rounding errors, a theoretical analysis indicates that the CG algorithm is able to

solve the discretized PDE exactly in $O(N)$ steps, using only $O(N^2)$ computation and $O(N)$ storage on a single processor. In practice, experience shows that the CG method, when applied to the PDE problem, usually converges in $O(\sqrt{N})$ steps even with rounding errors. Unfortunately, on a mesh-connected array, this algorithm is slowed by the need to compute several inner products of $O(N)$ length vector. Computing the inner product of two vectors whose entries are distributed over a mesh-connected array requires global communication. We therefore encounter the same difficulties as for the accelerated relaxation methods.

The local relaxation method proposed by Ehrlich [5], [6] and Botta and Veldman [3] is a computational algorithm suitable for parallel implementation on mesh-connected processor arrays, because it has the same acceleration effect as SOR and uses only local communication. A local relaxation procedure for equation (2.2) can be written as follows:

red points $(i+j$ *is even*$)$:

$$u_{i,j}^{(n+1)} = (1-\omega_{i,j})u_{i,j}^{(n)} + \omega_{i,j}d_{i,j}^{-1}(l_{i,j}u_{i-1,j}^{(n)} + r_{i,j}u_{i+1,j}^{(n)} + b_{i,j}u_{i,j-1}^{(n)} + t_{i,j}u_{i,j+1}^{(n)} + s_{i,j}). \tag{2.5a}$$

black points $(i+j$ *is odd*$)$:

$$u_{i,j}^{(n+1)} = (1-\omega_{i,j})u_{i,j}^{(n)} + \omega_{i,j}d_{i,j}^{-1}(l_{i,j}u_{i-1,j}^{(n+1)} + r_{i,j}u_{i+1,j}^{(n+1)} + b_{i,j}u_{i,j-1}^{(n+1)} + t_{i,j}u_{i,j+1}^{(n+1)} + s_{i,j}), \tag{2.5b}$$

where $\omega_{i,j}$ is called *the local relaxation parameter.*

Assuming Dirichlet boundary conditions and $M_1 \times M_2 = N$ unknowns within the unit square, it was suggested in [5], [6] that a good choice of local relaxation parameters $\omega_{i,j}$ is given by

$$\omega_{i,j} = \frac{2}{1+\sqrt{1-\rho_{i,j}^2}}, \tag{2.6}$$

where

$$\rho_{i,j} = \frac{2}{d_{i,j}}\left(\sqrt{l_{i,j}r_{i,j}}\cos\frac{\pi}{M_1+1} + \sqrt{t_{i,j}b_{i,j}}\cos\frac{\pi}{M_2+1}\right). \tag{2.7}$$

Since we consider only the case of symmetric discretized matrices, the parameter $\rho_{i,j}$ is always real. This gives us the *ad hoc SOR method* or the *local relaxation method* for a symmetric matrix. However, the local relaxation method can also be applied to more general matrices such that $\rho_{i,j}$ is purely imaginary or complex. The formula to determine the relaxation parameters for these cases can be found in [3], [5] and [6]. In this paper, we will focus on the local relaxation method for a system of equations $Au = s$ where A is symmetric positive definite. The more general case will be considered in a subsequent paper.

The implementation of the local relaxation method is straightforward. It is easy to see that as long as we know the size of the grid, i.e., M_1 and M_2, we can broadcast this information to all processors in the loading stage. Each processor has to compute its own relaxation parameters once according to equations (2.6) and (2.7); then the local iterations specified by (2.5) can be performed in parallel for all processors with only local communication. Since the local relaxation method uses local communication, the computation time per iteration is $O(1)$. We will show that the number of iterations for typical test problems is proportional to $O(\sqrt{N})$ in § 6. Therefore, the total running time becomes $O(\sqrt{N})$. For a $\sqrt{N}\times\sqrt{N}$ processor array, the constraint that each processor should contain a minimum amount of global information implies that the lower bound for the computation time for any algorithm is $O(\sqrt{N})$, since it takes

$O(\sqrt{N})$ time for the data at one edge of the array to move to the opposite edge. It turns out that the local relaxation method achieves this lower bound.

Although the local relaxation method was empirically shown to be powerful, there are several questions which were left unanswered by the papers of Ehrlich, Botta and Veldman. First, they did not prove that the local relaxation method converges. Furthermore, there was no explanation of why the local relaxation method converges very fast. In the following sections, we will explore these two issues.

3. Convergence of the local relaxation method. In this section, we give a sufficient condition for the convergence of a local relaxation procedure. Then, we show that the local relaxation method given by equations (2.5)-(2.7) indeed converges.

In order to obtain a convergence result which covers the most general type of local relaxation procedure, we use a matrix formulation, since such a formulation includes not only the 5-point stencil corresponding to the discretized equation (2.2), but also other kinds of stencils. Given a linear system of equations, $Au = s$, where A is an $N \times N$ real symmetric positive definite matrix with positive diagonal elements, we may rewrite A as

$$A = D - E - F = D(I - L - U) \quad \text{and} \quad E^T = F$$

where I, D, E and F represent identity, diagonal, lower and upper triangular matrices, and $L = D^{-1}E$ and $U = D^{-1}F$. Let W be the diagonal matrix formed by the local relaxation parameters, i.e., $W = \text{diag}\,(\omega_1, \omega_2, \cdots, \omega_N)$. Then, a local relaxation procedure can be written in matrix iterative form as

$$u^{(n+1)} = (I - WL)^{-1}[(I - W) + WU]u^{(n)} + (I - WL)^{-1}WD^{-1}s. \tag{3.1}$$

Let $\bar{u}$ be the solution of the above iterative equation, so that

$$\bar{u} = (I - WL)^{-1}[(I - W) + WU]\bar{u} + (I - WL)^{-1}WD^{-1}s.$$

Define the error vector at nth iteration as $e^{(n)} = u^{(n)} - \bar{u}$. Then the matrix iterative equation in the error space becomes

$$e^{(n+1)} = (I - WL)^{-1}[(I - W) + WU]e^{(n)}. \tag{3.2}$$

The iteration matrix of the local relaxation procedure (3.1) is therefore given by $G(W) = (I - WL)^{-1}[(I - W) + WU]$. The iteration procedure will converge for all initial estimates $\bar{u}^{(0)}$ if and only if all eigenvalues of $G(W)$ are less than one in magnitude, i.e., if the spectral radius $\rho[G(W)]$ of the iteration matrix $G(W)$ is less than 1. A simple sufficient condition for convergence is given by the following theorem.

THEOREM 1 (Sufficient condition for the convergence of a local relaxation procedure). *Suppose A is an $N \times N$ real symmetric positive definite matrix. For the local relaxation procedure given by* (3.1), *if $0 < \omega_i < 2$ for $1 \leqq i \leqq N$, then $\rho[G(W)] < 1$ and the iterative algorithm converges.*

Proof. Let λ and p be an arbitrary eigenvalue, eigenvector pair of $G(W)$. Then $G(W)p = \lambda p$, or equivalently,

$$[(I - W) + WU]p = \lambda(I - WL)p. \tag{3.3}$$

Premultiplying by $p^H DW^{-1}$ on both sides, we obtain

$$p^H DW^{-1}p - p^H Dp + p^H DUp = \lambda p^H DW^{-1}p - \lambda p^H DLp.$$

Since $E^T = F$, $E = DL$, and $F = DU$, it is easy to check that

$$p^H DUp = (Lp)^H Dp = \overline{p^H DLp}.$$

Defining $z = p^H DLp/p^H Dp$ and $1/\omega = p^H DW^{-1}p/p^H Dp$, (3.3) can be simplified as

$$\frac{1}{\omega} - 1 + \bar{z} = \frac{\lambda}{\omega} - \lambda z,$$

or equivalently,

$$\lambda = \frac{1 - \omega + \omega\bar{z}}{1 - \omega z}.$$

Let $z = r e^{j\theta}$, then

$$|\lambda|^2 = \lambda\bar{\lambda} = 1 - \frac{\omega(2-\omega)(1-2r\cos\theta)}{(1-\omega r\cos\theta)^2 + \omega^2 r^2 \sin^2\theta}. \tag{3.4}$$

We know that $|\lambda|^2$ is always positive. If we can show that the second term in the above expression is also positive, then we can conclude that $|\lambda|$ is less than 1. The denominator of the second term of equation (3.4) is positive, so that we only have to consider the numerator. We have

$$2r\cos\theta = 2\,\mathrm{Re}\,(z) = \bar{z} + z = \frac{p^H DLp}{p^H Dp} + \frac{p^H DUp}{p^H Dp} = 1 - \frac{p^H Ap}{p^H Dp} < 1,$$

where the inequality is due to the fact that A and D are both positive definite. Note that since A is positive definite, the matrix D formed with the diagonal elements of A is also positive definite. Therefore, we know that $1 - 2r\cos\theta > 0$. Now, consider the range of the parameter ω. Since $W = \mathrm{diag}\,(\omega_1, \omega_2, \cdots, \omega_N)$, $W^{-1} = \mathrm{diag}\,(\omega_1^{-1}, \omega_2^{-1}, \cdots, \omega_N^{-1})$. Assuming that all relaxation factors are positive, we have

$$\frac{1}{\omega_{\max}} < \frac{p^H DW^{-1}p}{p^H Dp} = \frac{\sum_{i=1}^N |p_i|^2 d_i \omega_i^{-1}}{\sum_{i=1}^N |p_i|^2 d_i} < \frac{1}{\omega_{\min}},$$

where $\omega_{\max}$ and $\omega_{\min}$ are the largest and smallest eigenvalues of the matrix W and p_i is the ith element of the vector p. If we set $0 < \omega_{\min} \leqq \omega_{\max} < 2$, then

$$0 < \omega_{\min} \leqq \omega \leqq \omega_{\max} < 2.$$

Under this condition, the second term in equation (3.4) is always positive, so that the eigenvalues of the matrix $G(W)$ are all less than 1 and the local relaxation procedure (3.1) converges. Q.E.D.

The above theorem gives the range of the local relaxation parameters which guarantees that a local relaxation procedure converges; however, it does not tell us how to choose the relaxation parameters to make a local relaxation procedure converge faster. The local relaxation method mentioned in the last section is a special case of a local relaxation procedure, where the local relaxation parameters are specified for a 5-point stencil discretization. To show its convergence, we only have to show that all relaxation parameters chosen by the rule (2.6), (2.7) are between 0 and 2.

COROLLARY (Convergence of the local relaxation method for a 5-point stencil discretization). *The local relaxation method for a 5-point stencil given by* (2.5)-(2.7) *converges.*

Proof. From the discussion in the previous section, we know that the matrix A obtained by discretizing (2.1) is symmetric positive definite.

Since $p(x_1, x_2)$ and $q(x_1, x_2)$ are positive functions and $\sigma(x_1, x_2)$ is a nonnegative function, we know from (2.3) that $l_{i,j}$, $r_{i,j}$, $b_{i,j}$, $t_{i,j}$ and $d_{i,j}$ are all positive. In addition,

$0 < \cos \pi/(M+1) < 1$ for $1 < M < \infty$. Therefore $\rho_{i,j}$ given by (2.7) is also positive. Using the inequalities

$$2\sqrt{l_{i,j}r_{i,j}} \leqq l_{i,j} + r_{i,j}, \qquad 2\sqrt{t_{i,j}b_{i,j}} \leqq t_{i,j} + b_{i,j},$$

we have

$$\rho_{i,j} = \frac{2}{d_{i,j}}\left(\sqrt{l_{i,j}r_{i,j}} \cos\frac{\pi}{M_1+1} + \sqrt{t_{i,j}b_{i,j}} \cos\frac{\pi}{M_2+1}\right)$$

$$\leqq \frac{l_{i,j}+r_{i,j}}{d_{i,j}} \cos\frac{\pi}{M_1+1} + \frac{t_{i,j}+b_{i,j}}{d_{i,j}} \cos\frac{\pi}{M_2+1} < \frac{l_{i,j}+r_{i,j}+t_{i,j}+b_{i,j}}{d_{i,j}} \leqq 1$$

where the last inequality is obtained by noting that $\sigma_{i,j} \geqq 0$ in (2.3b). It is easy to see that

$$0 < \omega_{i,j} = \frac{2}{1+\sqrt{1-\rho_{i,j}^2}} < 2$$

for $0 < \rho_{i,j} < 1$. The local relaxation parameters chosen by the local relaxation method satisfy the sufficient condition given in Theorem 1, so that the relaxation method converges. Q.E.D.

4. Fourier analysis of the local relaxation method—5-point stencil. The convergence rate of a local relaxation procedure depends on how we choose the local relaxation parameters. The conventional SOR method chooses a spatially invariant relaxation parameter $\omega_{i,j} = \omega$ to minimize the asymptotic convergence rate, or, equivalently, minimize the spectral radius of $G(W)$. Young [18] showed that the optimal choice for ω in the accelerated Gauss-Seidel iteration is

$$\omega = \frac{2}{1+\sqrt{1-\rho^2}}$$

where ρ is the spectral radius of $D^{-1}(E+F)$. For this relaxation parameter, all eigenvalues of $G(\omega I)$ can be shown to have magnitude $\omega - 1$. In practice, it is quite difficult to calculate ρ exactly, and thus adaptive procedures are required to estimate ρ as the computation proceeds. In this section, we will use a Fourier analysis approach to derive a simple formula for a spatially varying relaxation parameter. Our formula is identical to that suggested by Ehrlich [5]. Our approach demonstrates that this formula will indeed achieve an excellent convergence rate. This study also gives some new insight into Young's SOR method.

For a linear constant coefficient PDE with Dirichlet or periodic boundary conditions, the eigenfunctions of $D^{-1}(E+F)$ are sinusoidal functions. Therefore, the spectral radius of this iterative matrix can be obtained by using Fourier analysis. However, for a space-varying coefficient PDE with general boundary conditions, the sinusoidal functions are not eigenfunctions. As a consequence, Fourier analysis cannot be applied rigorously. Notwithstanding this disadvantage, Fourier analysis is still a convenient tool for understanding the convergence properties of relaxation methods [16]. A more rigorous treatment to make Fourier analysis applicable to space-varying coefficient PDEs with general boundary conditions is needed and is currently under study. Roughly speaking, the reason why Fourier analysis often works in spatially varying PDE problems is that the eigenfunctions can be regarded as sinusoidal functions plus some perturbations. As long as the perturbation is comparatively small, the sinusoidal function is a good approximation of the original eigenfunction. Therefore, Fourier analysis is still a good analytical tool. A detailed formulation of Fourier analysis in this general context will be presented elsewhere.

In § 4.1, we will show how to find the lowest Fourier component for given boundary conditions. Then, we use Fourier analysis to analyze the Jacobi relaxation method in § 4.2. This approach is sometimes called *the local Fourier analysis* [16]. Finally, we justify the efficiency of the local relaxation method. The derivation can be viewed as a generalization of Brandt's local Fourier analysis to the Successive Over-Relaxation case.

4.1. Admissible error function space and its lowest Fourier component. Let Γ_i, $1 \leqq i \leqq 4$ denote the four boundaries of the unit square. Consider a set of linear first-order boundary conditions such as (2.1b) on the boundaries of the unit square,

$$B_i u = g_i \quad \text{on } \Gamma_i, \quad 1 \leqq i \leqq 4 \tag{4.1}$$

where B_i represents the boundary condition operator on the ith boundary. It is more convenient to analyze the relaxation in the error space rather than in the solution space, because the error equations are homogeneous. The error formulation for the boundary conditions can be obtained as follows. Let $\bar{u}$ be the actual solution so that

$$B_i \bar{u} = g_i \quad \text{on } \Gamma_i, \quad 1 \leqq i \leqq 4. \tag{4.2}$$

Subtracting (4.2) from (4.1), we obtain the homogeneous PDE in the error,

$$B_i e = 0 \quad \text{on } \Gamma_i, \quad 1 \leqq i \leqq 4. \tag{4.3}$$

The functions defined on the unit square and satisfying the homogeneous boundary conditions (4.3) are called the *admissible error functions*, since any error function allowed in the relaxation process should always satisfy the given boundary conditions. All admissible error functions form the *admissible error function space.* The sinusoidal functions in the admissible error function space can be chosen as a basis of this space because of their completeness. As far as the convergence rate is concerned, we will see that only the lowest frequency component is relevant. Thus, we will find that only the lowest frequency of this basis needs to be determined.

We assume that all B_i's are constant-coefficient operators. Under this assumption, B_1 and B_3 are independent of the x_2-direction, B_2 and B_4 are independent of the x_1-direction, and since the problem domain is square, the admissible Fourier components can be written in separable form as $s_1(x_1)s_2(x_2)$, where $s_1(\cdot)$ and $s_2(\cdot)$ are two 1-D sinusoidal functions. The boundary condition on Γ_1 becomes

$$B_1 s_1(x_1) s_2(x_2) = s_2(x_2) B_1 s_1(x_1) = 0,$$

i.e.,

$$B_1 s_1(x_1) = 0.$$

Similarly, we simplify the boundary conditions on Γ_2, Γ_3 and Γ_4, and decompose the 2-D problem into two independent 1-D problems.

$$\text{(I)} \quad B_1 s_1(x_1) = 0 \text{ when } x_1 = 0, \qquad B_3 s_1(x_1) = 0 \text{ when } x_1 = 1, \tag{4.4a}$$

$$\text{(II)} \quad B_2 s_2(x_2) = 0 \text{ when } x_2 = 0, \qquad B_4 s_2(x_2) = 0 \text{ when } x_2 = 1. \tag{4.4b}$$

From (4.4a) and (4.4b), we can determine the lowest frequencies $\hat{k}_1$ and $\hat{k}_2$ separately. We only show how to get $\hat{k}_1$ from (I); then $\hat{k}_2$ can be obtained from (II) in the same way.

Consider the mixed type boundary operators,

$$B_1 = b_1 + b_2 \frac{d}{dx_1} \quad \text{for } x_1 = 0, \tag{4.5a}$$

$$B_3 = b_3 + b_4 \frac{d}{dx_1} \quad \text{for } x_1 = 1. \tag{4.5b}$$

The Fourier component $s(k_1, x_1)$ of $s(x_1)$ at the frequency k_1 can be written as a linear combination of two complex sinusoids $e^{ik_1x_1}$ and $e^{-ik_1x_1}$, i.e.,

$$s(k_1, x_1) = c(k_1)\, e^{ik_1x_1} + c(-k_1)\, e^{-ik_1x_1}. \tag{4.6}$$

Substituting (4.6) into (4.5), we obtain

$$(b_1 + ib_2k_1)c(k_1) + (b_1 - ib_2k_1)c(-k_1) = 0,$$

$$(b_3 + ib_4k_1)\, e^{ik_1x_1}c(k_1) + (b_3 - ib_4k_1)\, e^{-ik_1x_1}c(-k_1) = 0.$$

In order to get nonzero values for $c(k_1)$ and $c(-k_1)$, the determinant of the 2×2 coefficient matrix should equal zero, or equivalently,

$$e^{i2k_1} = \frac{(b_1 + ib_2k_1)(b_3 - ib_4k_1)}{(b_1 - ib_2k_1)(b_3 + ib_4k_1)}. \tag{4.7}$$

Therefore, we conclude that the frequency k_1 of any admissible 1-D sinusoidal function with respect to the boundary conditions (4.5) must satisfy equation (4.7).

Let us look at two examples. If the boundary conditions on both Γ_1 and Γ_3 are Dirichlet type boundary conditions, which means b_2 and b_4 are zeros, then (4.7) becomes

$$e^{i2k_1} = 1 \quad \text{or} \quad \cos 2k_1 + i \sin 2k_1 = 1.$$

The solutions are $k_1 = n\pi$, $n = 0, \pm 1, \pm 2, \cdots$. However, it is easy to see that the zero frequency cannot be allowed. Thus, the lowest Fourier frequency $\hat{k}_1$ in the admissible error space is π. If we change the boundary condition on Γ_3 to be of Neumann type, i.e., $b_3 = 0$ but $b_4 \neq 0$, then (4.7) becomes

$$e^{i2k_1} = -1 \quad \text{or} \quad \cos 2k_1 + i \sin 2k_1 = -1.$$

The solutions are $k_1 = \frac{1}{2}n\pi$, where n is odd and the lowest frequency $\hat{k}_1$ is $\pi/2$.

The same procedure applies to other boundary conditions. Notice that the determination of the lowest Fourier components of a given PDE requires only the knowledge of the boundary conditions and of the geometry of the problem domain. The above procedure does not require any information about the PDE operator.

4.2. Local Jacobi relaxation operator and its properties. In this section, we use a Fourier analysis approach to analyze the local Jacobi operator and to determine its largest eigenvalue, or spectral radius, for given boundary conditions as previously discussed. The spectral radius of a local Jacobi operator will be used to determine the optimal local relaxation parameter of the local relaxation scheme in § 4.3.

Define the x_1-*direction* (x_2-*direction*) *forward-shift* and *backward-shift operators*, E_1 and E_1^{-1} (E_2 and E_2^{-1}), as

$$E_1 u_{i,j} = u_{i+1,j}, \qquad E_1^{-1} u_{i,j} = u_{i-1,j},$$

$$E_2 u_{i,j} = u_{i,j+1}, \qquad E_2^{-1} u_{i,j} = u_{i,j-1}.$$

Then, the 5-point discretization formula for an interior grid point can be written as

$$L_{i,j}u_{i,j} = s_{i,j}$$

where $L_{i,j} \equiv d_{i,j} - (r_{i,j}E_1 + l_{i,j}E_1^{-1} + t_{i,j}E_2 + b_{i,j}E_2^{-1})$ is the *local discretized differential operator at node* (i, j). The Jacobi relaxation at a local node can be written as

$$u_{i,j}^{(n+1)} = J_{i,j}u_{i,j}^{(n)} + s_{i,j}, \qquad n \geqq 0,$$

where $J_{i,j} \equiv d_{i,j}^{-1}\ (r_{i,j}E_1 + l_{i,j}E_1^{-1} + t_{i,j}E_2 + b_{i,j}E_2^{-1})$ is the *local Jacobi relaxation operator.* From the error point of view, we get

$$e_{i,j}^{(n+1)} = J_{i,j}e_{i,j}^{(n)}, \qquad n \geqq 0.$$

If the input error function $e_{i,j}^{(n)}$ is the complex sinusoid $e^{i(k_1x_1+k_2x_2)}$, we have

$$J_{i,j}\, e^{i(k_1x_1+k_2x_2)} = \mu_{i,j}(k_1, k_2)\, e^{i(k_1x_1+k_2x_2)}$$

where $\mu_{i,j}(k_1, k_2) = d_{i,j}^{-1}\ (r_{i,j}\, e^{ik_1h} + l_{i,j}\, e^{-ik_1h} + t_{i,j}\, e^{ik_2h} + b_{i,j}\, e^{-ik_2h})$. Therefore, we may view $e^{i(k_1x_1+k_2x_2)}$ as an eigenfunction of $J_{i,j}$ with eigenvalue $\mu_{i,j}(k_1, k_2)$. The magnitude of $\mu_{i,j}(k_1, k_2)$ provides some information on how the errors of different frequency components are smoothed out by the Jacobi relaxation process. This quantity can be computed as

$$|\mu_{i,j}(k_1, k_2)| = \frac{([(r_{i,j} + l_{i,j}) \cos k_1h + (t_{i,j} + b_{i,j}) \cos k_2h]^2 + [(r_{i,j} - l_{i,j}) \sin k_1h + (t_{i,j} - b_{i,j}) \sin k_2h]^2)^{1/2}}{d_{i,j}}. \tag{4.8}$$

By assuming that the coefficient functions are smooth so that

$$r_{i,j} - l_{i,j} = l_{i+1,j} - l_{i,j} = O(h) \quad \text{and} \quad t_{i,j} - b_{i,j} = b_{i+1,j} - b_{i,j} = O(h),$$

then the two cosine terms in $|\mu_{i,j}|$ are the dominant terms.

The eigenvalue function $\mu_{i,j}(k_1, k_2)$ is usually called the *frequency response* in signal processing [11] and the Jacobi relaxation operator can be viewed as a filtering process in the frequency domain. The frequency response function with the magnitude shown in (4.8), in fact, represents a 2-D notch filter instead of a lowpass filter. However, if the discretization space h is small enough and the waveforms are band-limited, this is not a significant problem. The reason is best explained from the Taylor's series approximation of a function $f(x)$, i.e.,

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + \frac{(x - x_0)^2}{2!}f''(x_0) + \cdots.$$

Supposing $f(x) = e^{ikx}$ and $x = x_0 + h$, the high order terms are negligible only if the product kh is reasonably small, say, less than 1. That means that as long as the magnitude of wavevector k is bounded, we can always find a discretization spacing h which is fine enough so that the dimensionless frequencies $\theta_1 = k_1h$ and $\theta_2 = k_2h$ are always inside the unit circle in the (θ_1, θ_2) plane. In this region, the notch filter behaves like a lowpass filter. The lowpass filtering property makes the error at higher frequencies converge to zero faster than that at lower frequencies.

The eigenvalue with the largest magnitude is the dominant factor in the *asymptotic convergence rate analysis,* so that we will focus our attention on this quantity. Following the above discussion, we define the *spectral radius* $\rho_{i,j}$ of $J_{i,j}$ as the largest magnitude of $\mu_{i,j}(k_1, k_2)$, i.e.,

$$\rho_{i,j} \equiv \max_{k_1,k_2} |\mu_{i,j}(k_1, k_2)|.$$

For the symmetric positive definite matrix case, the magnitude of $\mu_{i,j}(k_1, k_2)$ is the largest at the lowest frequency $(\hat{k}_1, \hat{k}_2)$, since such a choice makes the dominant cosine terms of (4.8) as large as possible. Therefore, we obtain

$$\rho_{i,j} = |\mu_{i,j}(\hat{k}_1, \hat{k}_2)|.$$

The above procedure, known as local Fourier analysis of $J_{i,j}$, has two implicit assumptions. First, $J_{i,j}$ is space-invariant. Secondly, the problem domain should be either extended to infinity or be rectangular but with periodic boundary conditions. In general, these two assumptions do not hold. As a consequence, $\rho_{i,j}$ is a spatially varying function and is not equal to the spectral radius ρ of the original Jacobi relaxation matrix $J = D^{-1}(E+F)$. An important question to be answered is whether the knowledge of $\rho_{i,j}$ can provide us some information about ρ. Two observations may be of help. First, the *same* lowest frequency gives the spectral radii of all local Jacobi relaxation operators, so this frequency should play a role in determining the eigenfunction giving the spectral radius of the Jacobi relaxation matrix J. Furthermore, for a given low frequency $(\hat{k}_1, \hat{k}_2)$, $\rho_{i,j}$ is a very smooth function in space. It is neither sensitive to variations of the coefficient functions nor sensitive to changes in the boundary conditions. For example, the values of $\rho_{i,j}$ given by equations (2.7) and (4.8), computed for Dirichlet and periodic boundary conditions separately, are only slightly different under the assumption that the coefficient functions are smooth. Let $\bar{\rho} = \max_{i,j} \rho_{i,j}$ and $\underline{\rho} = \min_{i,j} \rho_{i,j}$. Then, ρ should be a quantity somewhere between $\underline{\rho}$ and $\bar{\rho}$. Usually, the difference between $\bar{\rho}$ and $\underline{\rho}$ is so small that *any* $\rho_{i,j}$ can give us an estimate of ρ.

Notice that in order to determine the spectral radius of a local relaxation operator, we only have to know the lowest admissible Fourier component corresponding to the given boundary conditions, discussed in § 4.1, and then to compute $\rho_{i,j}$ according to (4.8).

4.3. Applying Fourier analysis to the local relaxation method. Let us reconsider the local relaxation method, i.e., equation (2.5). We divide the problem domain into red and black points and update one color at each time step. Suppose we start with the relaxation of the red points, then with the black points. The local equations for the error can be written as

$$e_{i,j}^{(n+1)} = (1-\omega_{i,j})e_{i,j}^{(n)} + \omega_{i,j}J_{i,j}e_{i,j}^{(n)} \quad \text{for } i+j \text{ even}, \tag{4.9}$$

$$e_{i,j}^{(n+1)} = (1-\omega_{i,j})e_{i,j}^{(n)} + \omega_{i,j}J_{i,j}e_{i,j}^{(n+1)} \quad \text{for } i+j \text{ odd}. \tag{4.10}$$

If all $J_{i,j}$'s are approximately the same within that small region, then we can combine (4.9) with (4.10) and rewrite (4.10) as

$$e_{i,j}^{(n+1)} = (1-\omega_{i,j})e_{i,j}^{(n)} + \omega_{i,j}(1-\omega_{i,j})J_{i,j}e_{i,j}^{(n)} + \omega_{i,j}^2 J_{i,j}^2 e_{i,j}^{(n)} \quad \text{for } i+j \text{ odd}. \tag{4.11}$$

Let e_R and e_B represent respectively the errors at the red and black points around node (i,j). Notice that equations (4.9) and (4.11) describe, in fact, the relation of two waves—the *red* and *black* waves in the local region around node (i,j). Rearranging and simplifying (4.9) and (4.11), we obtain the following relation between two successive iterations,

$$\begin{pmatrix} e_R^{(n+1)} \\ e_B^{(n+1)} \end{pmatrix} = \begin{bmatrix} 1-\omega_{i,j} & \omega_{i,j}J_{i,j} \\ \omega_{i,j}(1-\omega_{i,j})J_{i,j} & 1-\omega_{i,j}+\omega_{i,j}^2 J_{i,j}^2 \end{bmatrix} \begin{pmatrix} e_R^{(n)} \\ e_B^{(n)} \end{pmatrix}, \qquad n \geqq 0,$$

where the 2×2 matrix operator $G_{i,j}(\omega_{i,j}, J_{i,j})$ appearing in the above equation is called *the local relaxation operator with relaxation factor* $\omega_{i,j}$ *at node* (i,j).

By assuming that an eigenfunction of the local relaxation operator $G_{i,j}$ has the form $(c_1 e^{i(k_1x_1+k_2x_2)}, c_2 e^{i(k_1x_1+k_2x_2)})^T$ and that the corresponding eigenvalue is $\lambda_{i,j}$, we may write

$$G_{i,j}(\omega_{i,j}, J_{i,j}) \begin{pmatrix} c_1 e^{i(k_1x_1+k_2x_2)} \\ c_2 e^{i(k_1x_1+k_2x_2)} \end{pmatrix} = \lambda_{i,j} \begin{pmatrix} c_1 e^{i(k_1x_1+k_2x_2)} \\ c_2 e^{i(k_1x_1+k_2x_2)} \end{pmatrix}. \tag{4.12}$$

Equation (4.12) can be further simplified because of the assumption that the local operator $J_{i,j}$ is approximately constant in the region around the node (i, j) and has the eigenvalue $\mu_{i,j}$ for this complex sinusoid $e^{i(k_1x_1+k_2x_2)}$. We obtain

$$G_{i,j}(\omega_{i,j}, \mu_{i,j})\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \lambda_{i,j}\begin{pmatrix} c_1 \\ c_2 \end{pmatrix}. \tag{4.13}$$

Note that the eigenvalues of the operator matrix $G_{i,j}(\omega_{i,j}, J_{i,j})$ are the same as those of the matrix $G_{i,j}(\omega_{i,j}, \mu_{i,j})$.

Furthermore, from (4.13), we know that $\mu_{i,j}(k_1, k_2)$ and $\lambda_{i,j}(k_1, k_2)$ are related via

$$|G_{i,j}(\omega_{i,j}, \mu_{i,j}) - \lambda_{i,j}I| = 0,$$

or, equivalently,

$$\lambda_{i,j}^2 - (2 - 2\omega_{i,j} + \omega_{i,j}^2\mu_{i,j}^2)\lambda_{i,j} + (1-\omega_{i,j})^2 = 0.$$

Therefore, we get

$$\lambda_{i,j} = 1 - \omega_{i,j} + \frac{\omega_{i,j}^2\mu_{i,j}^2}{2} \pm \frac{\sqrt{\Delta}}{2}, \tag{4.14a}$$

where

$$\Delta = 4(1-\omega_{i,j})\omega_{i,j}^2\mu_{i,j}^2 + \omega_{i,j}^4\mu_{i,j}^4. \tag{4.14b}$$

Let us consider the special case, $\omega_{i,j} = 1$, which corresponds to the Gauss-Seidel relaxation method. The eigenvectors of the 2×2 matrix $G_{i,j}(\omega_{i,j}, \mu_{i,j})$ are $(1, 0)^T$ and $(1, \mu_{i,j})^T$ and the corresponding eigenvalues are 0 and $\mu_{i,j}^2$. This means that if we start with two sinusoidal waveforms at the same frequency but with different amplitudes, one of them, the red wave represented by the vector (1, 0), disappears in one step. The other wave remains and alternates between the red and black points thereafter, as mentioned in § 2. The ratio between the updated wave and the old wave is equal to the constant $\mu_{i,j}$, so that the amplitude is reduced by a factor of $\mu_{i,j}^2$ per cycle.

The purpose of introducing the relaxation parameter $\omega_{i,j}$ is to make the eigenvalue $\lambda_{i,j}(k_1, k_2, \omega_{i,j})$ of the new operator $G_{i,j}$ smaller than the eigenvalue $\mu_{i,j}(k_1, k_2)$ of the old operator $J_{i,j}$. For a fixed real $\mu_{i,j}(k_1, k_2)$, the relationship between $\lambda_{i,j}$ and $\omega_{i,j}$ can be described by the root locus technique depicted in Fig. 3.

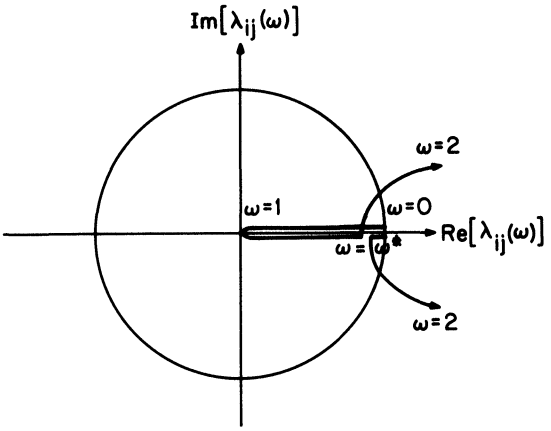


FIG. 3. *Root loci of* $\lambda_{i,j}(\omega)$ *with fixed* $\mu_{i,j}$.

When $0<\omega_{i,j}<2$, the magnitude of $\lambda_{i,j}$ is less than one. By Theorem 1, we know that if $0<\omega_{i,j}<2$ for all i,j then $\rho[G(W)]<1$ and the local relaxation algorithm converges. When $\Delta=0$, the two eigenvalues $\lambda_{i,j,1}$ and $\lambda_{i,j,2}$ coincide, and the largest possible magnitude of these two eigenvalues, $\lambda_{i,j,m}\equiv\max(|\lambda_{i,j,1}|,|\lambda_{i,j,2}|)$, is minimized. The value of $\omega_{i,j}$ which sets $\Delta=0$ is called the *optimal relaxation factor with respect to a specific* $\mu_{i,j}$ and is denoted by $\omega_{i,j,opt}(\mu_{i,j})$. By solving

$$\Delta=4(1-\omega_{i,j})\omega_{i,j}^2\mu_{i,j}^2+\omega_{i,j}^4\mu_{i,j}^4=0$$

and requiring

$$0<\omega_{i,j}<2,$$

we find that

$$\omega_{i,j,opt}(\mu_{i,j})=\frac{2}{1+\sqrt{1-\mu_{i,j}^2}}. \tag{4.15}$$

The general relation between $\lambda_{i,j,m}$ and $\omega_{i,j}$ can be derived in a straightforward way from equation (4.14) and is given by

$$\lambda_{i,j,m}=\omega_{i,j}-1, \qquad \omega_{i,j,opt}(\mu_{i,j})\leqq\omega_{i,j}<2, \tag{4.16a}$$

$$\lambda_{i,j,m}=\left[\frac{\omega_{i,j}|\mu_{i,j}|+\sqrt{\omega_{i,j}^2\mu_{i,j}^2+4(1-\omega_{i,j})}}{2}\right]^2, \qquad 0<\omega_{i,j}\leqq\omega_{i,j,opt}(\mu_{i,j}). \tag{4.16b}$$

The minimum value of all possible $\lambda_{i,j,m}$'s is, therefore, $\omega_{i,j,opt}(\mu_{i,j})-1$.

Since $\mu_{i,j}$ is a function of frequency, (4.15) implies that different frequencies require different optimal relaxation factors. However, we are allowed to choose only one $\omega_{i,j}$, so we have to consider the overall performance, i.e., $\omega_{i,j}$ has to be selected so that the spectral radius of the local relaxation operator $G_{i,j}(\omega_{i,j},J_{i,j})$ is minimized over all frequencies. Let $\rho_{i,j}$ be the spectral radius of the local operator $J_{i,j}$ and $\mu_{i,j}$ be an arbitrary eigenvalue of $J_{i,j}$. By definition, $|\mu_{i,j}|\leqq\rho_{i,j}$, so we know that $\omega_{i,j,opt}(\mu_{i,j})\leqq\omega_{i,j,opt}(\rho_{i,j})$ from (4.15). Using the relation in (4.16), we reason as follows. If we choose $\omega_{i,j}=\omega_{i,j,opt}(\mu_{i,j})$, $\lambda_{i,j,m}(\mu_{i,j})$ achieves its minimum value of $\omega_{i,j,opt}(\mu_{i,j})-1$ but $\lambda_{i,j,m}(\rho_{i,j})$ is greater than $\omega_{i,j,opt}(\rho_{i,j})-1$. On the other hand, if $\omega_{i,j,opt}(\rho_{i,j})$ is chosen as the relaxation factor, both $\lambda_{i,j,m}(\mu_{i,j})$ and $\lambda_{i,j,m}(\rho_{i,j})$ are equal to $\omega_{i,j,opt}(\rho_{i,j})-1$. Comparing these two cases, the latter choice is the best scheme to minimize the spectral radius of $G_{i,j}(\omega_{i,j},J_{i,j})$. This optimal value of $\omega_{i,j}$ is denoted as $\omega_{i,j}^*$, and is given by

$$\omega_{i,j}^*=\omega_{i,j,opt}(\rho_{i,j})=\frac{2}{1+\sqrt{1-\rho_{i,j}^2}}.$$

This is exactly the same formula as suggested by Ehrlich. The reason that this is a good choice is due to the fact that the eigenfunction with the largest eigenvalue of the Jacobi relaxation operator is the one corresponding to the lowest frequency component, and to the observation that the space varying relaxation parameter $\omega_{i,j}^*$ optimizes the convergence of this lowest frequency mode.

5. The local relaxation method for a 9-point stencil. The above derivation applies to a 5-point stencil, which appears when we discretize a linear second-order elliptic PDE without the crossover term $\partial^2/\partial x_1\,\partial x_2$. If there is a crossover term, a finite difference discretization gives a 9-point stencil. In this section, we will propose a local relaxation scheme for a 9-point stencil, give a sufficient condition for its convergence, and use a Fourier analysis approach to explain the rule for selecting good local relaxation parameters. The approach is similar to that used in §§ 3 and 4.

In order to make the presentation clear, we use a simple problem mentioned in [1] as an illustrative example. Consider the linear partial differential operator defined on $\Omega = [0, 1] \times [0, 1]$,

$$L = \frac{\partial^2}{\partial x_1^2} + a(x_1, x_2) \frac{\partial^2}{\partial x_1 \partial x_2} + \frac{\partial^2}{\partial x_2^2} \tag{5.1}$$

where $|a(x_1, x_2)| < 2$, with appropriate boundary conditions. The condition $|a(x_1, x_2)| < 2$ is required to guarantee that L is an elliptic operator. We also assume that $a(x_1, x_2)$ is sufficiently smooth so that it can be viewed as being approximately constant locally. The following discretization scheme is used

$$\frac{\partial^2}{\partial x_1^2} \leftrightarrow \frac{E_1 - 2 + E_1^{-1}}{h^2}, \qquad \frac{\partial^2}{\partial x_2^2} \leftrightarrow \frac{E_2 - 2 + E_2^{-1}}{h^2},$$

$$\frac{\partial^2}{\partial x_1 \partial x_2} \leftrightarrow \frac{E_1 E_2 + E_1^{-1} E_2^{-1} - E_1^{-1} E_2 - E_1 E_2^{-1}}{4h^2}.$$

The local Jacobi relaxation operator $J_{i,j}$ can be decomposed into two parts $J_{i,j,1}$ and $J_{i,j,2}$, i.e.,

$$J_{i,j} = J_{i,j,1} + J_{i,j,2},$$

where

$$J_{i,j,1} = \tfrac{1}{4}(E_1 + E_1^{-1} + E_2 + E_2^{-1}),$$

$$J_{i,j,2} = \tfrac{1}{16}(a_{i+1/2,j+1/2} E_1 E_2 + a_{i-1/2,j-1/2} E_1^{-1} E_2^{-1} - a_{i-1/2,j+1/2} E_1^{-1} E_2 - a_{i+1/2,j-1/2} E_1 E_2^{-1}),$$

and where

$$a_{i,j} = a(ih, jh).$$

Suppose we use the red/black partitioning; then the $J_{i,j,1}$ operator couples nodes of different colors while the $J_{i,j,2}$ operator couples nodes of the same color. A four-color scheme which leads to a four-color SOR method has been proposed for this problem [1]. Here, we propose a different local relaxation scheme which uses only red/black partitioning, so that the iteration equations for the error in the local region can be written as

$$e_R^{(n+1)} = (1 - \omega_{i,j}) e_R^{(n)} + \omega_{i,j} [J_{i,j,1} e_B^{(n)} + J_{i,j,2} e_R^{(n)}], \tag{5.2a}$$

$$e_B^{(n+1)} = (1 - \omega_{i,j}) e_B^{(n)} + \omega_{i,j} [J_{i,j,1} e_R^{(n+1)} + J_{i,j,2} e_B^{(n)}], \tag{5.2b}$$

or, equivalently,

$$\begin{pmatrix} e_R^{(n+1)} \\ e_B^{(n+1)} \end{pmatrix} = G_{i,j}(\omega_{i,j}, J_{i,j,1}, J_{i,j,2}) \begin{pmatrix} e_R^{(n)} \\ e_B^{(n)} \end{pmatrix}$$

where

$$G_{i,j}(\omega_{i,j}, J_{i,j,1}, J_{i,j,2}) = \begin{bmatrix} 1 - \omega_{i,j} + \omega_{i,j} J_{i,j,2} & \omega_{i,j} J_{i,j,1} \\ \omega_{i,j}(1 - \omega_{i,j}) J_{i,j,1} + \omega_{i,j}^2 J_{i,j,1} J_{i,j,2} & 1 - \omega_{i,j} + \omega_{i,j} J_{i,j,2} + \omega_{i,j}^2 J_{i,j,1}^2 \end{bmatrix}. \tag{5.3}$$

In general, the linear system of equations, $Au = s$ for a 9-point stencil which is obtained by discretizing an elliptic PDE with a crossover term can be decomposed as

$$Au = (D - E - F - C)u = D(I - L - U - V)u = s, \tag{5.4}$$

where A is an $N \times N$ real symmetric positive definite matrix and D, E, F and C are diagonal, lower and upper triangular, and block diagonal matrices respectively. In (5.4), we view the 9-point stencil as the superposition of a standard 5-point stencil and of a 4-point stencil formed by the nodes at the four corners. The standard 5-point stencil is accounted for by $D-E-F$, while the remaining 4-point stencil due to the crossover term is represented by the matrix $C = DV$. It is not hard to see that

$$E^T = F \quad \text{and} \quad C^T = C.$$

According to the local relaxation method specified by (5.2), the matrix iterative equation in the error space becomes

$$e^{(n+1)} = (I - WL)^{-1}[(I - W) + WU + WV]e^{(n)}, \tag{5.5}$$

where W is a diagonal matrix formed by local relaxation parameters. The iteration matrix is therefore given by

$$G(W) = (I - WL)^{-1}[(I - W) + WU + WV].$$

A simple sufficient condition for the convergence of (5.5), or (5.2), can be obtained by generalizing Theorem 1. Following the same steps as in the proof of Theorem 1, we find that

$$\lambda = \frac{1 - \omega(1-\alpha) + \omega\bar{z}}{1 - \omega z},$$

where λ, p is an arbitrary eigenvalue/eigenvector pair of $G(W)$, $z = p^H DLp / p^H Dp$, $1/\omega = p^H DW^{-1}p / p^H Dp$ and $\alpha = p^H Cp / p^H Dp$. Since C is symmetric, α is a real number. Let $z = r e^{j\theta}$; then

$$|\lambda|^2 = 1 - \frac{\omega[2 - \omega(1-\alpha)](1 - \alpha - 2r\cos\theta)}{(1 - \omega r\cos\theta)^2 + \omega^2 r^2 \sin^2\theta}, \tag{5.6}$$

which is similar to (3.4). Now, consider

$$\begin{aligned} \alpha + 2r\cos\theta &= \alpha + 2\,\mathrm{Re}\,(z) = \alpha + \bar{z} + z \\ &= \frac{p^H DVp}{p^H Dp} + \frac{p^H DLp}{p^H Dp} + \frac{p^H DUp}{p^H Dp} = 1 - \frac{p^H Ap}{p^H Dp} < 1, \end{aligned}$$

where the inequality is due to the fact that A and D are both positive definite. Furthermore, let us assume that $\alpha < 1$. In order to guarantee that $|\lambda| < 1$ for all possible eigenvalues, the sufficient condition becomes

$$0 < \omega_{\min} \leqq \omega_{i,j} \leqq \omega_{\max} < \frac{2}{1 - \alpha_{\min}},$$

where

$$\alpha_{\min} = \min_p \frac{p^H Cp}{p^H Dp}, \tag{5.7}$$

and where the minimization is over all eigenvectors p of the matrix $G(W)$. Therefore, we have the following theorem.

THEOREM 2 (Sufficient condition for the convergence of a local relaxation procedure for a 9-point stencil discretization). *Suppose that A is an $N \times N$ real symmetric positive definite matrix. For the local relaxation procedure given by* (5.4) *and the constant $\alpha_{\min}$ defined by* (5.7), *if $0 < \omega_i < 2/(1 - \alpha_{\min})$ for $1 \leqq i \leqq N$, then $\rho[G(W)] < 1$ and the iterative algorithm converges.*

Note that if there is no crossover term, the matrix C is zero and α is also zero. In this case, the 9-point stencil reduces to a 5-point stencil and Theorem 2 reduces to Theorem 1. Therefore, our proposed local relaxation scheme for a 9-point stencil, (5.5), is a natural generalization of the conventional SOR method for a 5-point stencil, specified by (3.2).

In the above derivation, we have used the assumption that $\alpha = p^H Cp / p^H Dp$ is less than 1 for any eigenvector p of the matrix $G(W)$. Now, let us estimate the value of α by examining the example given by (5.1). For simplicity, we consider the special case where $a(x_1, x_2) = a$ is constant and assume that the boundary conditions are *periodic*, i.e., $u(0, x_2) = u(1, x_2)$ for $0 \leqq x_2 \leqq 1$, $u(x_1, 0) = u(x_1, 1)$ for $0 \leqq x_1 \leqq 1$. In this case, the eigenvectors p of $G(W)$ can be found in closed form and are given by one of the following two-dimensional arrays, $\sin(k_1 ih + k_2 jh)$, $\cos(k_1 ih + k_2 jh)$, $\sin(k_1 ih - k_2 jh)$ and $\cos(k_1 ih - k_2 jh)$, where i and j range from 1 to $\sqrt{N}$, $h = 1/\sqrt{N}$, and k_1, k_2 are multiples of 2π. Then, after some computations, we find that

$$\alpha(p) = \frac{p^H Cp}{p^H Dp} = \pm \frac{a}{4} \sin k_1 h \sin k_2 h \quad \text{and} \quad |\alpha(p)| < \frac{|a|}{4} \quad \text{for } \textit{all } p.$$

Therefore, if we choose ω_i between 0 and $8/(4+|a|)$, the local relaxation algorithm for this particular problem will converge. However, this choice is too conservative to give a good convergence rate when $|a|$ is close to 2.

Generally speaking, two types of errors arise in the numerical solution of elliptic PDEs by iterative methods. The first of these is caused by the error between the initial guess and the true solution. The other is the numerical rounding error due to the finite precision arithmetic. The first error is usually concentrated in the low frequency region, whereas the second can exist at all possible frequencies. The numerical rounding error is usually so small that it can be ignored, provided it does not grow with the number of iterations. Thus, the error smoothing primarily aims at reducing errors in the low frequency region where the initial guess errors are substantial.

Let us temporarily ignore the numerical rounding error and focus on the initial guess error only. In order to guarantee the convergence of all components in the low frequency region, we need only to select

$$\alpha_{\min} := \alpha^L_{\min} \equiv -\frac{|a|}{4} \sin(\tilde{k}_1 h) \sin(\tilde{k}_2 h)$$

where $\tilde{k}_1$ and $\tilde{k}_2$ are the largest frequencies of interest. Usually the mesh is so fine that $\alpha^L_{\min}$ is of order $O(h^2)$. Although this conclusion is obtained from a simple example, it seems reasonable to believe that $\alpha^L_{\min}$ is also of order $O(h^2)$ for more general second-order elliptic PDEs with space-varying coefficients and other boundary conditions.

The remaining problem is to select a set of local relaxation parameters such that the iterative algorithm converges as quickly as possible in the low frequency region. We can use the Fourier analysis approach introduced in § 4 to analyze the local 2×2 matrix operator $G_{i,j}(\omega_{i,j}, J_{i,j,1}, J_{i,j,2})$ given by (5.3). Let $\mu_{i,j,1}(k_1, k_2)$, $\mu_{i,j,2}(k_1, k_2)$ and $\mu_{i,j}(k_1, k_2)$ be eigenvalues of $J_{i,j,1}$, $J_{i,j,2}$ and $J_{i,j}$ respectively. Following the procedure used before, we find that the optimal local relaxation factor for (5.1) is

$$\omega^*_{i,j} = \frac{2}{1 - \varepsilon_{i,j} + \sqrt{(1-\varepsilon_{i,j})^2 - \rho^2_{i,j}}} \tag{5.8}$$

where $\varepsilon_{i,j} = \mu_{i,j,2}(\hat{k}_1, \hat{k}_2)$, $\rho_{i,j} = \mu_{i,j,1}(\hat{k}_1, \hat{k}_2)$, and $(\hat{k}_1, \hat{k}_2)$ is the mode which maximizes

$\mu_{i,j} = \mu_{i,j,1} + \mu_{i,j,2}$. The spectral radius of the local relaxation operator $G_{i,j}$ is

$$\lambda_{i,j}(G_{i,j}(\omega_{i,j}^*, J_{i,j,1}, J_{i,j,2})) = \omega_{i,j}^*(1+\varepsilon_{i,j}) - 1. \tag{5.9}$$

Typical values for $\omega_{i,j}^*$ and $\lambda_{i,j}$ can be obtained by considering example (5.1) with Dirichlet boundary conditions, which is of more interest compared to the periodic boundary conditions in practice. Since $a(x_1, x_2)$ is smooth, we can approximate $J_{i,j,2}$ by

$$J_{i,j,2} \approx \frac{a_{i,j}}{16}(E_1E_2 + E_1^{-1}E_2^{-1} - E_1^{-1}E_2 - E_1E_2^{-1}).$$

The values of $\mu_{i,j,1}(k_1, k_2)$, $\mu_{i,j,2}(k_1, k_2)$ and $\mu_{i,j}(k_1, k_2)$ are

$$\mu_{i,j,1}(k_1, k_2) = \frac{1}{2}(\cos k_1h + \cos k_2h),$$

$$\mu_{i,j,2}(k_1, k_2) = \pm\frac{a_{i,j}}{8}[\cos(k_1+k_2)h - \cos(k_1-k_2)h] = \pm\frac{a_{i,j}}{4}\sin k_1h \sin k_2h$$

and

$$\mu_{i,j}(k_1, k_2) = \frac{1}{2}(\cos k_1h + \cos k_2h) \pm \frac{a_{i,j}}{4}\sin k_1h \sin k_2h.$$

In the low frequency region, $J_{i,j,1}$ is a lowpass filter, while $J_{i,j,2}$ is a highpass filter. $J_{i,j}$ is similar to $J_{i,j,1}$ in this region, because $\mu_{i,j,2}$ is almost zero for very low frequencies. Therefore, we can view $J_{i,j,2}$ as a *perturbation.* Thus, $J_{i,j}$ is a perturbed lowpass filter. Its spectral radius $\rho_{i,j}(J_{i,j})$ is determined by the lowest admissible frequency. For this particular problem, the lowest admissible frequency mode turns out to be $(\hat{k}_1, \hat{k}_2) = (\pi, \pi)$. Therefore,

$$\varepsilon_{i,j} = \frac{|a_{i,j}|}{4}\sin^2 \pi h, \qquad \rho_{i,j} = \cos \pi h.$$

Substituting these values back into (5.8) and (5.9), we find

$$\omega_{i,j}^* \approx \frac{2}{1+(\sqrt{2+|a_{i,j}|}/\sqrt{2})\pi h} \approx 2\left(1 - \frac{\sqrt{2+|a_{i,j}|}}{\sqrt{2}}\pi h\right), \tag{5.10a}$$

$$\lambda_{i,j} \approx 1 - \sqrt{4+2|a_{i,j}|}\,\pi h. \tag{5.10b}$$

In general, $\alpha_{\min}^L$ is of $O(h^2)$ so that for sufficiently small values of h, the above optimal relaxation parameters satisfy the sufficient condition of Theorem 2. Hence, the convergence of the local relaxation method in the low frequency region is guaranteed and the convergence rate can be estimated by examining the spectral radius of the local relaxation operator given by (5.10b).

Now let us go back to the effect of numerical rounding errors. The optimal relaxation parameters $\omega_{i,j}^*$ given by (5.10a) may be outside the convergence range defined by 0 and $2/(1-\alpha(p))$ for some eigenvectors p corresponding to high frequency error components. Therefore, we expect that the error in the high frequency region will grow. The error growth rate is problem dependent and can be analyzed by Fourier analysis. If the error growth rate is so slow that it does not effect the answer much, we can stick to a single set of optimal local relaxation parameters. On the other hand, if the error growth rate is relatively large, we may use two sets of local relaxation parameters. One set aims at reducing the low frequency error quickly and the other

set, formed by smaller values of $\omega_{i,j}$, is used to smooth the high frequency error once in a while so that the rounding errors do not accumulate. This mixed scheme should perform much better than a scheme using a single set of conservative local relaxation parameters. However, the optimal scheduling of these two sets of local relaxation parameters is still unknown. We believe that it depends on the problem to be solved. Some numerical experiments will be needed to gain a better understanding of the issue.

6. Performance analysis of the local relaxation algorithm.

6.1. Convergence rate analysis—linear constant coefficient PDEs. For a linear constant coefficient PDE defined on a unit square with Dirichlet boundary conditions, the spectral radii of all local Jacobi operators are the same, and thus all local relaxation parameters and the spectral radii of all local relaxation operators are the same, i.e., for all i, j

$$\rho_{i,j} = \rho, \qquad \omega_{i,j} = \omega, \qquad \lambda_{i,j} = \lambda. \tag{6.1}$$

In this case, the local relaxation method is the same as SOR.

The *asymptotic convergence rate* of an arbitrary global iterative operator P, denoted by $R_\infty(P)$, is defined as [17]

$$R_\infty(P) = -\ln \rho(P).$$

Under the conditions (6.1), the asymptotic convergence rate is also given by

$$R_\infty(P) = -\ln \rho(P_{i,j})$$

where $P_{i,j}$ is the local relaxation operator of P.

For a Poisson equation on the unit square with Dirichlet boundary conditions, the local Jacobi operator for this particular problem is

$$J_{i,j} = \tfrac{1}{4}(E_1 + E_1^{-1} + E_2 + E_2^{-1}).$$

Applying Fourier analysis to $J_{i,j}$, we find that the spectral radius of $J_{i,j}$ is $\cos \pi h$, where h is the grid spacing. The global asymptotic convergence rate of the Jacobi method, the Gauss-Seidel method and the local relaxation method can be computed as

$$R_\infty(\text{Jacobi}) = -\ln \rho(J_{i,j}) = -\ln \cos \pi h \approx \frac{1}{2}\pi^2 h^2,$$

$$R_\infty(\text{Gauss-Seidel}) = -\ln \lambda[G_{i,j}(1, J_{i,j})] = -\ln \cos^2 \pi h \approx \pi^2 h^2,$$

$$R_\infty(\text{local relaxation}) = -\ln \lambda[(G_{i,j}(\omega^*, J_{i,j})] = -\ln\left(\frac{1-\sin \pi h}{1+\sin \pi h}\right) \approx 2\pi h.$$

Therefore, the number of iterations is proportional to $O(1/h)$, i.e., $O(\sqrt{N})$, for the local relaxation method.

6.2. Computer simulation. For general space-varying coefficient PDEs, it is difficult to analyze the convergence rate as shown in § 6.1. So, a simple numerical example is used to illustrate the convergence rate of the local relaxation method for solving space-varying coefficient PDEs. The convergence rates of the SOR and CG methods are also shown for the purpose of comparison. For the SOR and CG methods, the Ellpack software package [13] was used.

The example chosen is

$$e^{xy}\frac{\partial^2 u}{\partial x^2}+e^{-xy}\frac{\partial^2 u}{\partial y^2}+e^{xy}y\frac{\partial u}{\partial x}-e^{-xy}x\frac{\partial u}{\partial y}+\frac{1}{1+x+y}u$$

$$= e^{2xy}\sin \pi y[(2y^2-\pi^2)\sin \pi x+3\pi y\cos \pi x] \tag{6.2}$$

$$+\pi \sin \pi x(x\cos \pi y-\pi \sin \pi y)+\frac{e^{xy}\sin \pi x\sin \pi y}{1+x+y}$$

on the unit square, with the boundary conditions

$$u(x, y)=0 \quad \text{for } x=0,\ x=1,\ y=0 \text{ and } y=1, \tag{6.3}$$

and its solution is $e^{xy}\sin \pi x \sin \pi y$. Although equation (6.2) does not have the same form as (2.1), it is easy to verify that the discretized matrix is still symmetric positive definite so that Theorem 1 applies here. The lowest frequency mode for this problem is $(\hat{k}_1, \hat{k}_2)=(\pi, \pi)$ because of the Dirichlet boundary conditions.

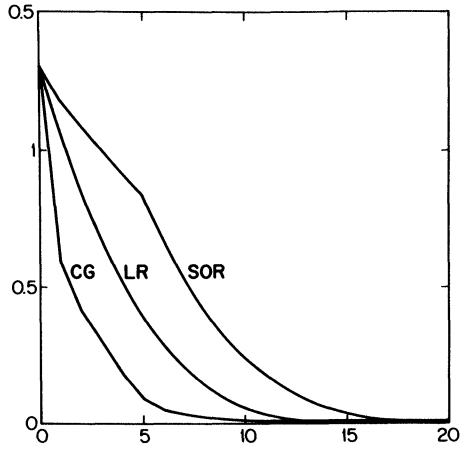
For this test problem, three 5-point discretization schemes are used with grid spacings 1/10, 1/30 and 1/50. Starting from the initial guess $u^{(0)}(x, y)=0$ for all grid points, the maximum errors at each iteration are plotted in Fig. 4. The results indicate that on a single processor the convergence rate of the local relaxation method is better than that of the SOR method and worse than that of the CG method. However, as mentioned in § 2, on a mesh-connected array the local relaxation takes constant time for each iteration while the CG and SOR methods take $O(\sqrt{N})$ time per iteration, so that the local relaxation method is much faster. We also note that the number of iterations required for the local relaxation method is proportional to $\sqrt{N}$. This is consistent with the analysis of the previous section.

7. Extensions and conclusions. The local relaxation method includes two important steps. The first is to determine the admissible lowest frequencies using boundary condition information. The second is to approximate the PDE operator locally by a linear finite difference operator, divide the nodes into red and black points and form a locally accelerated successive over-relaxation (local relaxation) operator. In previous discussions, some ideal assumptions were made so that the analysis and design of the local relaxation algorithm become very simple. However, we may encounter several difficulties in applying the local relaxation method directly to real world problems.

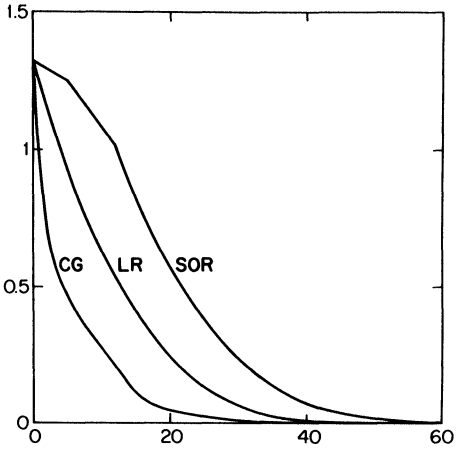
Under the assumption that the problem domain is a unit square and that the boundary condition operator is constant along each edge, the procedure for determining the lowest admissible frequencies is straightforward. These assumptions make the basis functions separable and easy to analyze. However, in practice, the above assumptions may not hold. The problem domain is usually of irregular shape and the boundary condition operators may have space-varying coefficients. As a consequence, it is considerably more difficult to find the lowest frequency error component than for the case we have considered in this paper.

The second difficulty is related to the construction of the local relaxation operator. If the coefficients of a PDE operator have some discontinuities in some region, the Jacobi relaxation operator is not smooth over the region with discontinuous coefficients. In this case, the determination of the optimal local relaxation factors for such abruptly changing operators is still an open question.

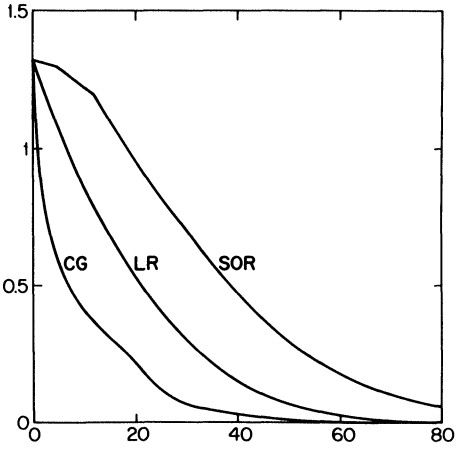
To map the irregular domain problem into a regular processor domain is important in practice. In addition, we need some schemes to partition the grid points evenly between all processors, when the number of grid points is larger than that of processors.



(a)



(b)



(c)

FIG. 4. *Computer simulation results for problem* (6.2)–(6.3) *with* (a) 11×11, (b) 31×31 *and* (c) 51×51 *grids.*

The mapping and partitioning problems have been studied recently [2], [7], but not too many results are known.

Other open questions include how to terminate the local relaxation method. A centralized termination scheme can work as follows. The residues of the local processors are pipelined to a certain processor, say, the central one. Then this processor determines the termination time of the relaxation algorithm and broadcasts a "stop" signal to all processors. The above procedure can be performed in parallel with the local relaxation procedures. Note however that because we avoid the use of global communications in the termination procedure by pipelining the local residues which are sent to the central processor, this processor uses the error residues of the local processors at *different* iterations in determining whether the algorithm should be stopped. A distributed termination scheme would also be useful.

It also seems interesting and challenging to see whether the local relaxation concept can be applied to elliptic PDEs which do not satisfy the assumptions made in § 2, such as the Helmholtz equation, or to other types of PDEs such as hyperbolic PDEs or nonlinear PDEs, and to other discretization schemes such as the finite-element method.

We may note that *distributed* computational PDE algorithms (local relaxation) are different from traditional *central* computational methods (SOR) in several ways. First, distributed computation provides a natural way to achieve a high degree of parallelism. Secondly, distributed algorithms suggest a space-adaptive acceleration scheme, which is not as convenient in centralized computation. Thirdly, although global information is required in determining the local optimal acceleration factors, it seems that only very little global information is relevant, namely the size of the domain. Finally, we benefit a lot in designing the local acceleration algorithm from the *simple* structure of the local operator and from the fact that the global information required is minimized, while the determination of the optimal uniform SOR acceleration factor is complicated and time-consuming.

These nice properties are closely related to the special structure of PDEs. Partial differential equations are formulated to describe local interactions in the physical world, where any interaction between two far space points is the result of a chain of local interactions between near space points. The locality property is very similar to the local communication constraint imposed by VLSI computation [10]. Therefore, although this constraint is a bottleneck in other types of problems, it is not as restrictive for numerical PDE problems.

Our paper has also demonstrated the use of the Fourier analysis approach, or frequency domain approach, to analyze the local relaxation algorithm. This methodology forms a bridge between numerical analysis for solving PDEs and digital signal processing [9]. This new method seems more informative than traditional matrix iterative methods, which usually hide information in a huge matrix. In addition, the local Fourier analysis approach provides a way to analyze distributed numerical algorithms while matrix iterative methods only can be applied to central numerical algorithms. A closer relationship between numerical analysis techniques and Fourier analysis is expected in the future.

Acknowledgments. The authors are grateful to Doctors Louis W. Ehrlich and Youcef Saad, and Professors Lloyd N. Trefethen and John Tsitsiklis for helpful discussions. Some comments from the reviewers were highly appreciated. The authors also wish to thank Professor William T. Thompkins, Jr. for his encouragement and support.

REFERENCES

[1] L. ADAMS AND J. M. ORTEGA, *A Multi-Color SOR Method for Parallel Computation*, ICASE report, 82-9, (1982).

[2] S. H. BOKHARI, *On the mapping problem*, IEEE Trans. Comput., C-30, 3 (1981), pp. 207-214.

[3] E. F. BOTTA AND A. E. P. VELDMAN, *On local relaxation methods and their application to convection-diffusion equations*, J. Comput. Phys., 48 (1981), pp. 127-149.

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

[5] L. W. EHRLICH, *An Ad Hoc* SOR *Method*, J. Comput. Phys., 44 (1981), pp. 31-45.

[6] ———, *The ad-hoc* SOR *method: a local relaxation scheme*, in Elliptic Problem Solvers II, Academic Press, New York, 1984.

[7] D. GANNON, *On mapping non-uniform PDE structures and algorithms onto uniform array architectures*, in Proc. International Conference on Parallel Processing, 1981.

[8] L. A. HAGEMAN AND D. M. YOUNG, *Applied Iterative Methods*, Academic Press, New York, 1981.

[9] C.-C. KUO, *Parallel algorithms and architectures for solving elliptic partial differential equations*, MIT Technical Report (LIDS-TH-1432), Cambridge, MA, January 1985.

[10] C. A. MEAD AND L. A. CONWAY, *Introduction to* VLSI *Systems*, Addison-Wesley, Reading, MA, 1980.

[11] A. V. OPPENHEIM AND R. W. SCHAFER, *Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1975.

[12] D. W. PEACEMAN, *Fundamentals of Numerical Reservoir Simulation*, Elsevier North-Holland, New York, 1977.

[13] J. R. RICE, ELLPACK: *Progress and Plans*, in Elliptic Problem Solvers, M. H. Schultz, ed., Academic Press, New York, 1981.

[14] P. J. ROACHE, *Computational Fluid Dynamics*, Hermosa Publishers, Albuquerque, NM, 1976.

[15] H. L. STONE, *Iterative solution of implicit approximations of multidimensional partial differential equations*, SIAM J. Numer. Anal., 5 (1958), pp. 530-558.

[16] K. STUBEN AND U. TROTTENBERG, *Multigrid methods: Fundamental algorithms, model problem analysis and applications*, in Multigrid Methods, U. Trottenberg, ed., Springer-Verlag, New York, 1982.

[17] R. S. VARGA, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1962.

[18] D. M. YOUNG, *Iterative methods for solving partial differential equations of elliptic type*, Ph.D. thesis, Harvard University, Cambridge, MA, 1950.