PARALLEL LINEAR STATIONARY ITERATIVE METHODS*

L. RIDGWAY SCOTT† AND DEXUAN XIE‡

Abstract. A parallel linear stationary iterative method, defined by domain partitioning and referred to as the JSOR method, is analyzed in this paper. Basic JSOR convergence theorems, including one concerning the optimal relaxation parameter, are presented. JSOR is shown to have a much faster convergence rate than Jacobi and the same efficiency of interprocessor-data communication as Jacobi. Since JSOR contains the classic SOR and damped-Jacobi methods as its two extreme cases, the JSOR analysis can lead to a general linear stationary iteration theory, and imply both SOR and damped-Jacobi theories directly. Numerical results are presented to demonstrate the parallel performance of JSOR on a MIMD parallel computer. Finally, the development and application of JSOR are discussed.

Key words. SOR, damped-Jacobi, JSOR, parallel computing, parallel multigrid methods.

AMS(MOS) subject classifications. 65F10, 65Y05.

1. Introduction.

The Jacobi and Gauss-Seidel methods (or damped-Jacobi and SOR for their relaxed variants) are two well known linear stationary iterations for solving linear systems of equations [10]. The Jacobi method is "completely" parallel, while the typically more efficient Gauss-Seidel method is the opposite. However, even though Jacobi can be implemented in parallel on n processors, where n is the number of unknowns of the linear system to be solved, the number p of processors actually employed is usually very small in comparison to n due to communication costs. In fact, on today's distributed memory MIMD architectures [3], the time required to update an iterative value is much smaller than the time required to send a updated iterative value to another processor, which may cause interprocessor data-communication overhead [1,3]. To overcome this overhead, a domain partitioning (or an index set partitioning for a general linear system) technique is often used in the implementation of Jacobi. That is, the unknowns of the linear system are divided into p groups corresponding to p subdomains of the domain partitioning, such that each group contains a sufficiently large and similar number of unknowns. Since each group is assigned to one processor, the iterative values within each group can be calculated sequentially. Hence, to improve the convergence rate of Jacobi, it is natural to substitute the Jacobi iterative process within each group by the sequential Gauss-Seidel iterative process, yielding a new par-

* This work was supported in part by NSF grant DMS-9105437.

† Department of Mathematics, University of Houston, Houston, TX 77204. Current address: Department of Mathematics, University of Chicago, Chicago, IL 60637 (ridg@uchicago.edu).

‡ Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012 (dexuan@cims.nyu.edu).

allel iterative method. We call this new scheme the JGS method because it is a mixture of Jacobi and Gauss-Seidel. Further, introducing a relaxation parameter to JGS, we obtain an accelerated variant of JGS, and call it the JSOR method. Obviously, JGS is JSOR with a unit relaxation parameter.

In this paper, we study the convergence properties and parallel performance of JSOR theoretically and numerically. We prove several basic convergence theorems for JSOR, including one concerning the optimal relaxation parameter. We also present a model problem analysis for JSOR to confirm our JSOR theory. In particular, we show that both damped-Jacobi and SOR are two extreme cases of JSOR (corresponding to $p = 1$ and $p = n$, respectively), and their convergence theorems directly follow from the corresponding JSOR convergence theorems. In this sense, the JSOR analysis can lead to a general linear stationary iteration theory.

JSOR has a faster convergence rate than Jacobi and the same efficiency of interprocessor data communication as Jacobi. We also observe that JGS can be regarded as a satisfactory parallel version of the sequential Gauss-Seidel method if p is not too large. However, in the case of optimal relaxation parameter, numerical results show that the convergence speed of JSOR can be reduced almost linearly with respect to p. Hence, JSOR cannot be taken as a parallel version of SOR; otherwise, no speedup can be obtained. Due to this, JSOR with $p > 1$ is not a suitable parallel solver of linear systems. This might explain why JSOR has not been studied so far even though the idea of JSOR has been known for a long time.

However, two motivations suggest a study of JSOR. First, we attempt to develop an efficient parallel SOR version by domain partitioning. Since JSOR is the simplest scheme by domain partitioning, a study of JSOR may help us to do so. It was based on our JSOR analysis that the PSOR method, an efficient parallel SOR method by domain partitioning, was obtained in [7]. As shown in [7], PSOR has the same convergence rate as the Red/Black SOR method (a widely-used parallel version of SOR) and the same advantages as JSOR in the parallel implementation. Hence, PSOR can be more efficiently and more easily implemented on a MIMD parallel computer than the Red/Black SOR method, especially for solving complicated scientific problems (for which it may be difficult to find a global Red/Black ordering).

The second motivation for us to analyze JSOR is to study the smoothing properties of JSOR. In fact, an important application of JSOR is as a parallel smoother of the parallel multigrid method [2,5]. Many numerical results have shown that JSOR can have the same smoothing effects as the SOR smoother when p is not too large [8,9]. However, before studying the smoothing properties of JSOR, we need to study JSOR mathematically. It was based on our JSOR analysis that we obtained a JSOR smoothing analysis in [9].

The remainder of this paper is organized as follows. In Section 2, we describe the JGS and JSOR methods. In Section 3, we analyze the conver-

gence of JSOR. In Section 4, we estimate the optimal relaxation parameter of JSOR. In Section 5, we give JSOR a model problem analysis. In Section 6, we demonstrate the parallel performance of JSOR. Conclusions are summarized in Section 7.

2. The JGS and JSOR methods. We consider the solution of the linear system

$$Au = f, \tag{2.1}$$

where $A = (a_{ij})_{n\times n}$ is a $n \times n$ nonsingular matrix, $u = (u_1, u_2, \ldots, u_n)^T$ is an unknown real vector, and $f = (f_1, f_2, \ldots, f_n)^T$ is a given real vector. The superscript T denotes a vector or matrix transpose.

Let $W = \{1, 2, \ldots, n\}$ be an index set. For a given positive integer p, we define a simple partition:[1]

$$W = R_1 \cup R_2 \cup \cdots \cup R_p \tag{2.2}$$

with $R_i = \{n_{i-1}+1, n_{i-1}+2, \ldots, n_i\}$ and n_i (for $i = 1, 2, \ldots, p$) satisfying

$$0 = n_0 < n_1 < n_2 < \cdots < n_p = n.$$

With this index set partition, we write A, u and f in the block forms

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1p} \\ A_{21} & A_{22} & \cdots & A_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ A_{p1} & A_{p2} & \cdots & A_{pp} \end{bmatrix}, \; u = \begin{bmatrix} \mathcal{U}_1 \\ \mathcal{U}_2 \\ \vdots \\ \mathcal{U}_p \end{bmatrix}, \text{ and } \; f = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_p \end{bmatrix}$$

such that (2.1) is written into a system of block equations

$$A_{ii}\mathcal{U}_i + \sum_{j=1, j\neq i}^{p} A_{ij}\mathcal{U}_j = F_i \quad \text{for } i = 1, 2, \ldots, p, \tag{2.3}$$

where the submatrix A_{ij} for $i, j = 1, 2, \ldots, p$ is defined from A by deleting all rows except those corresponding to indices in R_i and all columns vectors except those corresponding to indices in R_j, and $\mathcal{U}_i$ and F_i are formed from u and f by deleting all elements of u and f except those corresponding to indices in R_i respectively.

We first define the JGS iterates $u^k = (\mathcal{U}_1^{(k)}, \mathcal{U}_2^{(k)}, \cdots, \mathcal{U}_p^{(k)})^T$ by induction on k, where $k = 0, 1, 2, \ldots$. Let u^0 be a given initial guess. Assuming that u^k is known, with (2.3), we construct p independent block equations:

[1] If the linear system (2.1) arises from an approximation of an elliptic boundary problem on a grid mesh Ω_h, we partition Ω_h into p disjoint sub-meshes $\Omega_{h,i}$, and set $R_i = \{\ j \mid \text{mesh point } j \in \Omega_{h,i}\}$ for $i = 1, 2, \ldots, p$, so that JSOR is defined on the domain partitioning. With the appropriate numbering of grid points, these partitions coincide.

$$A_{ii}\mathcal{U}_i = F_i - \sum_{j=1, j\neq i}^{p} A_{ij}\mathcal{U}_j^{(k)}, \quad i = 1, 2, \ldots, p. \tag{2.4}$$

We then define the update $\mathcal{U}_i^{(k+1)}$ as the result of one step of Gauss-Seidel iteration for solving (2.4) with $\mathcal{U}_i^{(k)}$ as the starting point for each $i = 1, 2, \ldots, p$, yielding the $k+1$-th JGS iterate

$$u^{k+1} = (\mathcal{U}_1^{(k+1)}, \mathcal{U}_2^{(k+1)}, \ldots, \mathcal{U}_p^{(k+1)})^T.$$

Clearly, each $\mathcal{U}_i^{(k+1)}$ can be calculated independently. Hence, by assigning the calculation of $\mathcal{U}_i^{(k+1)}$ to processor i for $i = 1, 2, \ldots, p$, JGS can be implemented on p processors in parallel. After all $\mathcal{U}_i^{(k+1)}$ are calculated, the update $\mathcal{U}_i^{(k+1)}$ is sent from processor i to other processors as needed. Hence, each JGS iteration needs only one interprocessor data communication.

To get an iterative expression of JGS, we write A and submatrix A_{ii} $(i = 1, 2, \ldots, p)$ into the matrix sums

$$D^{-1}A = I - L - U, \quad \text{and} \quad (D_i)^{-1}A_{ii} = I_i - L_i - U_i, \tag{2.5}$$

where, respectively, D and D_i are the diagonal matrices of A and A_{ii}, I and I_i are two identity matrices, and L and U, as well as L_i and U_i, are strictly lower and strictly upper triangular matrices. We then write L and U in the following matrix sums:

$$L = B + N, \quad \text{and} \quad U = C + M, \tag{2.6}$$

where $B = \text{diag}(L_1, L_2, \ldots, L_p)$, $\quad C = \text{diag}(U_1, U_2, \ldots, U_p)$,

$$N = -D^{-1}\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ A_{21} & 0 & 0 & \cdots & 0 \\ A_{31} & A_{32} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ A_{p1} & A_{p2} & \cdots & A_{p,p-1} & 0 \end{bmatrix},$$

and

$$M = -D^{-1}\begin{bmatrix} 0 & A_{12} & A_{13} & \cdots & A_{1p} \\ 0 & 0 & A_{23} & \cdots & A_{2p} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & A_{p-1,p} \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}.$$

(The sum form of U, $U = C + M$, will be used only for the convergence analysis.)

By using this notation, the JGS iterates $\{u^k\}$ can be expressed by

$$u^{k+1} = M_p u^k + (I-B)^{-1}D^{-1}f, \quad k = 0, 1, 2, \ldots, \tag{2.7}$$

where the JGS iteration matrix M_p is below:

$$M_p = (I-B)^{-1}(U+N), \tag{2.8}$$

and the subscript p indicates that the iteration matrix depends on p.

We next define JSOR as an accelerated variant of JGS. That is, using the JGS iterate $\tilde{u} = Bu^{k+1} + (U+N)u^k + D^{-1}f$ and a given relaxation parameter ω, we define the JSOR iterates $\{u^k\}$ by

$$u^{k+1} = (1-\omega)u^k + \omega\tilde{u}, \quad k = 0, 1, 2, \ldots. \tag{2.9}$$

From (2.9) follows the iterative expression of JSOR

$$u^{k+1} = M_p(\omega)u^k + \omega(I-\omega B)^{-1}D^{-1}f, \quad k = 0, 1, 2, \ldots. \tag{2.10}$$

where the JSOR iteration matrix $M_p(\omega)$ has the expression

$$M_p(\omega) = (I-\omega B)^{-1}[(1-\omega)I + \omega(U+N)]. \tag{2.11}$$

Clearly, setting $\omega = 1$ in (2.11) gives the JGS iteration matrix M_p. Hence, JGS is a special case of JSOR.

JSOR has different iterative expressions with different values of p. In particular, JSOR with $p = 1$ and $p = n$ reduce to the well-known SOR and damped-Jacobi methods, respectively. In fact, for $p = 1$, we have $N = 0$ and $B = L$, so that $M_p(\omega) = (I-\omega L)^{-1}[(1-\omega)I + \omega U]$, which is the SOR iteration matrix [10]. Similarly, for $p = n$, $M_p(\omega) = (1-\omega)I + \omega(U + L)$, which is the damped-Jacobi iteration matrix. Fig. 1 depicts the two dimensional space of algorithms represented by JSOR and these various limits.

3. The convergence analysis. It is well known that a linear stationary iteration is convergent if and only if the spectral radius of the iterative matrix is less than one [10]. Hence, we will consider the spectral radius of the JSOR iteration matrix to study the convergence of JSOR. We denote the spectral radius as $\rho(\cdot)$.

THEOREM 3.1 (Comparison of the convergence rates of JGS, Gauss-Seidel, and Jacobi). *Let the matrix $A = (a_{ij})_{n\times n}$ satisfy $a_{ij} \le 0$ for all $i \ne j$ and $a_{ii} > 0$ for all i.*

(a) If $\rho(M_J) < 1$, then $\rho(M_p) \le \rho(M_J)$ for $1 \le p \le n$.

(b) If $\rho(M_J) \ge 1$, then $\rho(M_p) \ge \rho(M_J)$ for $1 \le p \le n$.

(c) If $\rho(M_J) < 1$ and $p > 1$, then $\rho(M_{GS}) < \rho(M_p) < 1$.

Here M_J, M_{GS}, and M_p are the iteration matrices of Jacobi, Gauss-Seidel, and JGS, respectively.

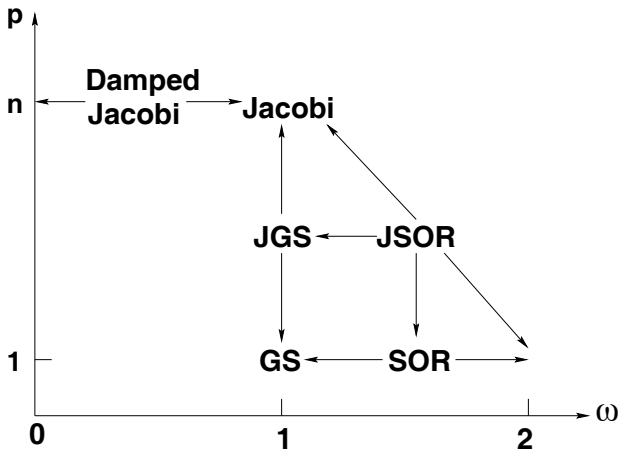

FIG. 1. *JSOR family of stationary iterative methods.*

Proof. Define the weighted maximum norm $\|\cdot\|_\infty^\xi$ by

$$\|x\|_\infty^\xi = \max_i \frac{|x_i|}{\xi_i},$$

where ξ is a given vector with its all components $\xi_i > 0$, and x_i is the i-th component of vector x. The matrix norm induced by $\|\cdot\|_\infty^\xi$ is defined by

$$\|A\|_\infty^\xi = \max_{\|x\|_\infty^\xi = 1} \|Ax\|_\infty^\xi, \tag{3.1}$$

which can be written in an equivalent form

$$\|A\|_\infty^\xi = \max_i \frac{\sum_{j=1}^n |a_{ij}|\xi_j}{\xi_i}. \tag{3.2}$$

(a) The Jacobi iteration matrix $M_J = (m_{ij})_{n\times n}$ has $m_{ij} = -a_{ij}/a_{ii}$ for $i \neq j$ and $m_{ii} = 0$ for all i. The assumption on A implies that M_J is a nonnegative matrix. Since $\rho(M_J) < 1$, for any $\epsilon > 0$ sufficiently small, $\lambda = \rho(M_J) + \epsilon < 1$. The Perron-Frobenius Theorem[2] [3] then implies that there exists some vector $\xi > 0$ such that $\|M_J\|_\infty^\xi \leq \lambda$, by (3.2), which can be written

$$\sum_{j=1}^n m_{ij}\xi_j \leq \lambda\xi_i, \quad i = 1, 2, \ldots, n. \tag{3.3}$$

[2] Some conclusions [3] from the Perron-Frobenius Theorem are as follows: *Let M be an $n \times n$ nonnegative matrix. Then there exists some nonzero vector $\xi \geq 0$ such that $M\xi = \rho(M)\xi$. Further, for every $\epsilon > 0$, there exists some $\xi > 0$ such that $\rho(M) \leq \|M\|_\infty^\xi \leq \rho(M) + \epsilon$.*

For the given ξ, we consider some x such that $\|x\|_\infty^\xi = 1$. This implies

$$|x_i| \le \xi_i \quad \text{for } i = 1, 2, \ldots, n. \tag{3.4}$$

Let $y = M_p x$. By (2.8), we have $y = By + Nx + Ux$ with the following component form

$$y_i = \sum_{j<i, j\in R_k} m_{ij} y_j + \sum_{j<i, j\notin R_k} m_{ij} x_j + \sum_{j>i} m_{ij} x_j, \quad i \in R_k, \tag{3.5}$$

where $k = 1, 2, \ldots, p$. Since $R_k = \{n_{k-1} + 1, \ldots, n_k\}$, it is sufficient to prove that $|y_i| \le \lambda \xi_i$ for all $i = 1, 2, \ldots, n$ by showing that $|y_i| \le \lambda \xi_i$ for all $i \in R_k$, where $k = 1, 2, \ldots, p$. We will do it by induction on i in each R_k. First, with (3.3) and (3.4),

$$\begin{aligned} |y_{n_{k-1}+1}| &= \left| \sum_{j=1}^{n} m_{(n_{k-1}+1)j} x_j \right| \\ &\le \sum_{j=1}^{n} m_{(n_{k-1}+1)j} |x_j| \le \sum_{j=1}^{n} m_{(n_{k-1}+1)j} \xi_j \le \lambda \xi_{n_{k-1}+1}. \end{aligned}$$

Assume that $|y_j| \le \lambda \xi_j$ for $j < i$, where $i, j \in R_k$. Then by (3.3), (3.4) and (3.5),

$$\begin{aligned} |y_i| &\le \sum_{j<i, j\in R_k} m_{ij} |y_j| + \sum_{j<i, j\notin R_k} m_{ij} |x_j| + \sum_{j>i} m_{ij} |x_j| \\ &\le \sum_{j<i, j\in R_k} m_{ij} \lambda \xi_j + \sum_{j<i, j\notin R_k} m_{ij} \xi_j + \sum_{j>i} m_{ij} \xi_j \\ &\le \sum_{j<i} m_{ij} \xi_j + \sum_{j>i} m_{ij} \xi_j \\ &= \sum_{j=1}^{n} m_{ij} \xi_j \le \lambda \xi_i, \end{aligned}$$

where $\lambda < 1$ has been used. Hence, by induction, we have proved that $|y_i| \le \lambda \xi_i$ for all $i \in R_k$ $(k = 1, 2, \ldots, p)$. Therefore,

$$\|M_p x\|_\infty^\xi = \|y\|_\infty^\xi = \max_{1\le i\le n} \frac{|y_i|}{\xi_i} = \max_{1\le k\le p} \max_{i\in R_k} \frac{|y_i|}{\xi_i} \le \lambda$$

for all x satisfying $\|x\|_\infty^\xi = 1$. This implies that

$$\rho(M_p) \le \|M_p\|_\infty^\xi \le \rho(M_J) + \epsilon.$$

Letting $\epsilon \to 0$, we complete the proof of (a).

(b) Since M_J is non-negative, by the Perron-Frobenius Theorem [3], there exists a nonzero vector $\xi \ge 0$ such that $M_J \xi = \rho(M_J)\xi$. Let $y = M_p \xi$.

Then $y \geq 0$. Noting that $\rho(M_J) \geq 1$, we can proceed as in the proof of part (a), with the inequalities reversed, to prove that $y_i \geq \rho(M_J)\xi_i$ for all $i \in R_k$, where $k = 1, 2, \ldots, p$. Therefore, we conclude that $M_p\xi = y \geq \rho(M_J)\xi$. For all integers $m > 0$, we then have $(M_p)^m\xi \geq \xi(\rho(M_J))^m$. Since $\xi \neq 0$, the matrix $(M_p/\rho(M_J))^m$ does not converge to zero as m tends to infinity. Therefore, $\rho(M_p/\rho(M_J)) \geq 1$, which gives $\rho(M_p) \geq \rho(M_J)$.

(c) Clearly, $M_{GS} = (I - L)^{-1}U$ is a nonnegative matrix. By the Perron-Frobenius Theorem [3], there exists a nonzero vector $\xi \geq 0$ such that $M_{GS}\xi = \rho(M_{GS})\xi$. This expression can be written as $\rho(M_{GS})\xi = U\xi + \rho(M_{GS})L\xi$ with the following component form

$$\rho(M_{GS})\xi_i = \sum_{j>i} m_{ij}\xi_j + \rho(M_{GS}) \sum_{j<i} m_{ij}\xi_j, \quad i = 1, 2, \ldots, n. \tag{3.6}$$

Further, according to part (a), $\rho(M_J) < 1$ follows $\rho(M_{GS}) < 1$.

Let $y = M_p\xi$. We will prove, by induction on i, that $y_i \geq \rho(M_{GS})\xi_i$ for all $i \in R_k$. Indeed, by (3.6) and $1 > \rho(M_{GS})$,

$$\begin{aligned} y_{(n_{k-1}+1)} &= \sum_{j=1}^{n} m_{(n_{k-1}+1)j}\xi_j \\ &\geq \sum_{j>n_{k-1}+1} m_{(n_{k-1}+1)j}\xi_j + \rho(M_{GS}) \sum_{j<n_{k-1}+1} m_{(n_{k-1}+1)j}\xi_j \\ &= \rho(M_{GS})\xi_{(n_{k-1}+1)}. \end{aligned}$$

Assuming that $y_j \geq \rho(M_{GS})\xi_j$ for $j < i$, where $i, j \in R_k$, we obtain

$$\begin{aligned} y_i &= \sum_{j<i, j\in R_k} m_{ij}y_j + \sum_{j<i, j\notin R_k} m_{ij}\xi_j + \sum_{j>i} m_{ij}\xi_j \\ &\geq \sum_{j<i, j\in R_k} m_{ij}\rho(M_{GS})\xi_j + \rho(M_{GS}) \sum_{j<i, j\notin R_k} m_{ij}\xi_j + \sum_{j>i} m_{ij}\xi_j \\ &= \rho(M_{GS}) \sum_{j<i} m_{ij}\xi_j + \sum_{j>i} m_{ij}\xi_j \\ &= \rho(M_{GS})\xi_i. \end{aligned}$$

This shows that

$$M_p\xi = y \geq \rho(M_{GS})\xi,$$

and from which we can prove that $\rho(M_p) \geq \rho(M_{GS})$ using the same arguments as in (b). ∎

Theorem 3.1 shows that JGS and Jacobi are either both convergent or both divergent. Asymptotically, all JGS iteration methods with $1 < p < n$ converge faster than Jacobi but slower than Gauss-Seidel. Since Gauss-Seidel is a particular case of JGS, the corresponding theorem about Gauss-Seidel (see Theorem 3.3 in [6]) follows from Theorem 3.1 immediately by setting $p = 1$.

THEOREM 3.2 (BASIC JSOR CONVERGENCE THEOREM). *Let A be a symmetric positive definite matrix, and define*

$$\tilde{\eta} = \min_{x \in \Lambda} \frac{x^* D(M+N)x}{x^* Dx}, \tag{3.7}$$

where Λ contains all nonzero eigenvectors of the JSOR iteration matrix $M_p(\omega)$, and the superscript * indicates the conjugate transpose. If $\tilde{\eta} < 1$, then JSOR is convergent if and only if $0 < \omega < \dfrac{2}{1-\tilde{\eta}}$.

Proof. Let x be a nonzero eigenvector of the JSOR iteration matrix $M_p(\omega)$, and λ be its corresponding eigenvalue such that $M_p(\omega)x = \lambda x$. Then, by (2.11), we have

$$[(1-\omega)I + \omega(U+N)]x = \lambda(I - \omega B)x. \tag{3.8}$$

Multiplying by x^*D on the both sides of (3.8), we get

$$\lambda = \frac{(1-\omega)x^*Dx + \omega(x^*DUx + x^*DNx)}{x^*Dx - \omega x^*DBx}. \tag{3.9}$$

Set $x^*DLx = \alpha + i\beta$, $x^*DNx = r_1 + ir_2$, and $x^*Dx = q$, where α, β, r_1, and r_2 are real, and $i = \sqrt{-1}$. From (2.5) we see that DU and DL are respectively strictly lower and strictly upper triangular matrices, whose entries are the negatives of the entries of A. Thus, the symmetry of A gives $DU = (DL)^T$ and $DM = (DN)^T$, so that $x^*DUx = \alpha - i\beta$, and $x^*DMx = r_1 - ir_2$. With $B = L - N$,

$$x^*DBx = x^*DLx - x^*DNx = (\alpha - r_1) + i(\beta - r_2).$$

Hence, (3.9) can be written as

$$\lambda = \frac{(1-\omega)q + \omega[(\alpha + r_1) - i(\beta - r_2)]}{q - \omega[(\alpha - r_1) + i(\beta - r_2)]}.$$

Since

$$|\lambda|^2 = \frac{[(1-\omega)q + \omega(\alpha + r_1)]^2 + \omega^2(\beta - r_2)^2}{[q - \omega(\alpha - r_1)]^2 + \omega^2(\beta - r_2)^2},$$

we get that $|\lambda| < 1$ if and only if

$$\omega(q - 2\alpha)q\left(2 - \omega + \frac{2r_1}{q}\right) > 0. \tag{3.10}$$

Here (3.10) is obtained from $[(1-\omega)q + \omega(\alpha + r_1)]^2 < [q - \omega(\alpha - r_1)]^2$. Further, the positive definiteness of A follows that

$$q = x^*Dx > 0 \quad \text{and} \quad q - 2\alpha = x^*Ax > 0. \tag{3.11}$$

Hence, (3.10) is equivalent to $2 - \omega + \dfrac{2r_1}{q} > 0$. Therefore, we have proved that

$$|\lambda| < 1 \quad \text{if and only if} \quad 2 - \omega + \omega\frac{2r_1}{q} > 0. \tag{3.12}$$

Suppose $0 < \omega < \dfrac{2}{1-\tilde{\eta}}$. This gives $2 - \omega(1-\tilde{\eta}) > 0$ since $\tilde{\eta} < 1$. By (3.7), we have

$$\tilde{\eta} \le \frac{x^* D(M+N)x}{x^* Dx} = \frac{2r_1}{q}.$$

Thus,

$$2 - \omega + \omega\frac{2r_1}{q} \ge 2 - \omega + \omega\tilde{\eta} = 2 - \omega(1-\tilde{\eta}) > 0. \tag{3.13}$$

Therefore, according to (3.12), we conclude $\rho(M_p(\omega)) < 1$.

Conversely, suppose $\rho(M_p(\omega)) < 1$. Then from (3.12) it follows

$$2 - \omega + \omega\frac{x^* D(M+N)x}{x^* Dx} > 0 \quad \text{for all } x.$$

Hence, by (3.7), we get $2 - \omega + \omega\tilde{\eta} > 0$, which implies $\omega < \dfrac{2}{1-\tilde{\eta}}$. □

THEOREM 3.3 (SUFFICIENT CONVERGENCE CONDITION FOR JSOR). *Let η be the smallest eigenvalue of the matrix $M + N$. Then JSOR is convergent for $0 < \omega < \dfrac{2}{1-\eta}$. Here $\eta < 0$ for $p > 1$, and $\eta = 0$ for $p = 1$.*

Proof. We first show that η can be expressed by

$$\eta = \min_{x \ne 0} \frac{x^* D(M+N)x}{x^* Dx}. \tag{3.14}$$

$M + N$ has the same eigenvalues as the matrix $D^{\frac{1}{2}}(M+N)D^{-\frac{1}{2}}$ because of their similarity. Since

$$D^{\frac{1}{2}}(M+N)D^{-\frac{1}{2}} = D^{-\frac{1}{2}}D(M+N)D^{-\frac{1}{2}} = D^{-\frac{1}{2}}(DM+DN)D^{-\frac{1}{2}},$$

and the symmetry of A gives $DM = (DN)^t$, we conclude that $D^{\frac{1}{2}}(M+N)D^{-\frac{1}{2}}$ is symmetric. Hence,

$$\eta = \min_{y \ne 0} \frac{y^* D^{\frac{1}{2}}(M+N)D^{-\frac{1}{2}}y}{y^* y} = \min_{x \ne 0} \frac{x^* D(M+N)x}{x^* Dx},$$

where we have set $y = D^{\frac{1}{2}}x$. This completes the proof of (3.14).

Since the trace of matrix $M+N$ is zero, the smallest eigenvalue η must be a negative real except the case of $p=1$. In fact, for $p=1$, we have $M=N=0$, following $\eta=0$. Further, from (3.14) it follows that $\eta \leq \tilde{\eta}$. Hence, when $0<\omega<\dfrac{2}{1-\eta}$, with (3.13), we have

$$2-\omega+\omega\frac{2r_1}{q} \geq 2-\omega(1-\eta) > 0.$$

Consequently, from (3.12) it follows that $\rho(M_p(\omega))<1$, i.e., JSOR is convergent. □

Since SOR and damped-Jacobi are JSOR using $p=1$ and n, respectively, setting $p=1$ and n in Theorem 3.2 immediately yields their well known convergence theorems (see Theorems 3.3 and 3.6 of Chapter 4 in [10]).

4. On the optimal relaxation parameter of JSOR. We say that ω_b is the optimal relaxation parameter of JSOR provided that it satisfies

$$\rho(M_p(\omega_b)) = \min_{\omega} \rho(M_p(\omega)). \tag{4.1}$$

Following the SOR optimal relaxation parameter theory [10], we will estimate the optimal relaxation parameter of JSOR in this section.

We first cite the definition of a consistently ordered matrix from [10].

DEFINITION 4.1. *The matrix $A=(a_{ij})_{n\times n}$ is consistently ordered if for some m there exist disjoint subsets $S_1, S_2, \ldots, S_m$ of $W=\{1,2,\ldots,n\}$ such that $\cup_{i=1}^{m} S_i = W$ and such that if $a_{i,j}\neq 0$ or $a_{j,i}\neq 0$, and $i\in S_k$, then $j\in S_{k+1}$ if $j>i$ and $j\in S_{k-1}$ if $j<i$.*

THEOREM 4.1 (Extension of Theorem 3.3 of Chapter 5 in [10]). *If A is a consistently ordered matrix, then for all reals β and κ, the determinant*

$$\Delta = \det(\alpha B + \alpha^{-1}C + \beta(N+M) - \kappa I)$$

is independent of α for all $\alpha\neq 0$. Here B, N, C and M are defined in (2.6).

Proof. Let $\sigma=(\sigma(1),\sigma(2),\ldots,\sigma(n))$ be a permutation defined on the integers $1,2,\ldots,n$. Based on the index set partition (2.2), we set

$$T_B = \{(i,j)\in R_k\times R_k \mid j<i, 1\leq k\leq p\},$$

$$T_C = \{(i,j)\in R_k\times R_k \mid j>i, 1\leq k\leq p\},$$

and $T_{N+M} = W\times W - T_B - T_C$. Since

$$\Delta = \frac{\det(D)\Delta}{\det(D)} = \frac{1}{\det(D)}\det(\alpha DB + \alpha^{-1}DC + \beta D(M+N) - \kappa D),$$

the general term of Δ is

$$t(\sigma) = \pm\frac{1}{\det(D)}\prod_{i=1}^{n} a_{i,\sigma(i)}\alpha^{l-\mu}\beta^{\nu}\kappa^{n-(l+\mu+\nu)},$$

where l, μ and ν are, respectively, the number of values of i such that $(i, \sigma(i)) \in T_B$, such that $(i, \sigma(i)) \in T_C$, and such that $(i, \sigma(i)) \in T_{N+M}$. Since $t(\sigma) = 0$ if and only if there exists one $a_{i\sigma(i)} = 0$, we only need to consider the terms with $a_{i\sigma(i)} \neq 0$ for $1 \leq i \leq n$.

Let l_k and μ_k be respectively the number of values of i such that $i > \sigma(i)$ and such that $i < \sigma(i)$ as well as such that $a_{i\sigma(i)}$ is an entry of A_{kk}. Clearly, A_{kk} is also consistently ordered. Hence, by the similar arguments in the proof of Theorem 3.3 in [10] we claim that $l_k = \mu_k$ for all $k = 1, 2, \ldots, p$. Noting that $D - DB - DC = \mathrm{diag}(A_{11}, \ldots, A_{pp})$, we have

$$l = \sum_{k=1}^{p} l_k \quad \text{and} \quad \mu = \sum_{k=1}^{p} \mu_k.$$

Hence, $l - \mu = \sum_{k=1}^{p} (l_k - \mu_k) = 0$. This proves that $t(\sigma)$ is independent of α. ∎

LEMMA 4.1. *Let A be a consistently ordered matrix with all its diagonal elements $a_{ii} \neq 0$. If λ is a nonzero eigenvalue of $M_p(\omega)$, then there exists nonzero real vector x such that*

$$\alpha + \lambda^{-\frac{1}{2}}\beta - \frac{\lambda + \omega - 1}{\omega\lambda^{\frac{1}{2}}} = 0, \tag{4.2}$$

where α and β are two real numbers defined by

$$\alpha = \frac{x^* D(C+B)x}{x^* Dx} \quad \text{and} \quad \beta = \frac{x^* D(M+N)x}{x^* Dx}. \tag{4.3}$$

Proof. Clearly, $\det(I - \omega B) = 1$, and $(I - \omega B)M_p(\omega) = (1-\omega)I + \omega(U+N)$. From Theorem 4.1 it follows that

$$\begin{aligned}
&\det\left(M_p(\omega) - \lambda I\right) \\
&\quad = \det(I - \omega B)\det\left(M_p(\omega) - \lambda I\right) \\
&\quad = \det\left(\omega(U+N) + \lambda\omega B - (\lambda + \omega - 1)I\right) \\
&\quad = \det\left(\omega\lambda^{\frac{1}{2}}\left[\lambda^{-\frac{1}{2}}(U+N) + \lambda^{\frac{1}{2}}B - \frac{\lambda+\omega-1}{\omega\lambda^{\frac{1}{2}}}I\right]\right) \qquad (4.4) \\
&\quad = \left(\omega\lambda^{\frac{1}{2}}\right)^n \det\left(\lambda^{-\frac{1}{2}}(U+N) + \lambda^{\frac{1}{2}}B - \frac{\lambda+\omega-1}{\omega\lambda^{\frac{1}{2}}}I\right) \\
&\quad = \left(\omega\lambda^{\frac{1}{2}}\right)^n \det\left(\lambda^{-\frac{1}{2}}C + \lambda^{\frac{1}{2}}B + \lambda^{-\frac{1}{2}}(M+N) - \frac{\lambda+\omega-1}{\omega\lambda^{\frac{1}{2}}}I\right) \\
&\quad = \left(\omega\lambda^{\frac{1}{2}}\right)^n \det\left(C + B + \lambda^{-\frac{1}{2}}(M+N) - \frac{\lambda+\omega-1}{\omega\lambda^{\frac{1}{2}}}I\right).
\end{aligned}$$

If λ is a nonzero eigenvalue of $M_p(\omega)$, (4.4) gives

$$\det\left(C + B + \lambda^{-\frac{1}{2}}(M+N) - \frac{\lambda+\omega-1}{\omega\lambda^{\frac{1}{2}}}I\right) = 0.$$

It then follows that there exists a nonzero real vector x such that

$$\left(C + B + \lambda^{-\frac{1}{2}}(M+N) - \frac{\lambda+\omega-1}{\omega\lambda^{\frac{1}{2}}}I\right)x = 0. \tag{4.5}$$

By setting α and β in the form of (4.3), (4.2) follows from (4.5). □

As an application of an algebra lemma[3] on the roots of a quadratic equation in [10], we can get the following lemma.

LEMMA 4.2. *Let λ be a root of the quadratic equation*

$$\lambda^2 + \lambda[2(\omega - \omega\beta - 1) - (\omega\alpha)^2] + [\omega(1-\beta) - 1]^2 = 0. \tag{4.6}$$

If $\beta < 1$, then $|\lambda| < 1$ if and only if

$$0 < \omega < \frac{2}{1-\beta} \quad \text{and} \quad |\alpha| < 1 - \beta, \tag{4.7}$$

where α and β are defined in (4.3), and (4.6) is a simplified form of Equation (4.2).

LEMMA 4.3. *The two roots λ_1 and λ_2 of Equation (4.6) satisfy $|\lambda_1| \geq |\lambda_2|$ for all α, β, and $\omega > 0$. In particular,*

$$|\lambda_1| = \begin{cases} \frac{1}{4}\left[\omega|\alpha| + \sqrt{\omega^2\alpha^2 - 4(\omega - \omega\beta - 1)}\right]^2 & \text{if } E \geq 0, \\ \omega - \omega\beta - 1 & \text{if } E < 0, \end{cases} \tag{4.8}$$

where $E = \omega^2\alpha^2 - 4(\omega - \omega\beta - 1)$.

Proof. It is easy to find the two roots of (4.6) as follows:

$$\lambda_1 = \frac{1}{4}\left[\omega|\alpha| + \sqrt{\omega^2\alpha^2 - 4(\omega - \omega\beta - 1)}\right]^2, \tag{4.9}$$

$$\lambda_2 = \frac{1}{4}\left[\omega|\alpha| - \sqrt{\omega^2\alpha^2 - 4(\omega - \omega\beta - 1)}\right]^2. \tag{4.10}$$

Obviously, $|\lambda_1| = |\lambda_2|$ if $E = 0$. If $E > 0$, then λ_1 and λ_2 are two positive real numbers, and $\lambda_1 - \lambda_2 = \omega|\alpha|\sqrt{E} > 0$. If $E < 0$, λ_1 and λ_2 are two conjugate complex numbers, we have $|\lambda_1| = |\lambda_2|$. Hence, for all α, β and $\omega > 0$, we have $|\lambda_1| \geq |\lambda_2|$.

We next show that $|\lambda_1|$ can be expressed by (4.8). If $E \geq 0$, then λ_1 is positive, so that $|\lambda_1| = \lambda_1 = \frac{1}{4}\left[\omega|\alpha| + \sqrt{\omega^2\alpha^2 - 4(\omega - \omega\beta - 1)}\right]^2$.

If $E < 0$, then λ_1 is a complex number with modulus $|\lambda_1| = \omega - \omega\beta - 1$. This completes the proof of (4.8). □

[3] The algebra lemma on the roots of a quadratic equation is as follows: *Let x_1 and x_2 be the two roots of a quadratic equation $x^2 - bx + c = 0$. Then $|x_1| < 1$ and $|x_2| < 1$ if and only if $|c| < 1$ and $|b| < 1 + c$.*

Due to Lemma 4.3, the estimation of $\rho(M_p(\omega))$ can be done by only considering λ_1.

Since it is complicated to express $\rho(M_p(\omega))$ and ω_b explicitly, we estimate $\rho(M_p(\omega))$ by an upper bound in Theorem 4.2, along with an approximation of ω_b in Theorem 4.3. The upper bound is sharp because it can be achieved by two particular JSOR algorithms using $p = 1$ and n (i.e., damped-Jacobi and SOR) as shown in Corollaries 4.2 and 4.3.

THEOREM 4.2. *Let $\underline{\beta}$ and $\hat{\beta}$ be the smallest and largest eigenvalues of $M + N$, respectively. If A is a consistently ordered symmetric matrix with all its diagonal elements $a_{ii} \neq 0$, $\rho(M_J) < 1$, and*

$$0 < \omega < \frac{2}{1 - \underline{\beta}}, \tag{4.11}$$

then $\rho(M_p(\omega)) < 1$ (i.e., JSOR is convergent), and for $E \geq 0$,

$$\begin{aligned} &\rho(M_p(\omega)) \\ &\quad \leq \frac{1}{4}\left[\omega(\rho(M_J) - \hat{\beta}) + \sqrt{\omega^2(\rho(M_J) - \hat{\beta})^2 - 4(\omega - \omega\hat{\beta} - 1)}\right]^2, \end{aligned} \tag{4.12}$$

while for $E < 0$,

$$\rho(M_p(\omega)) \leq \omega(1 - \underline{\beta}) - 1, \tag{4.13}$$

where $E = \omega^2\alpha^2 - 4(\omega - \omega\beta - 1)$.

Proof. Obviously, we have $\underline{\beta} \leq \beta \leq \hat{\beta}$ with the β given in (4.3). When $E \geq 0$, according to Lemma 4.3, we can set an upper bound of $\rho(M_p(\omega))$ as the maximum value of the function

$$\lambda(\alpha, \beta) = \frac{1}{4}\left[\omega|\alpha| + \sqrt{\omega^2\alpha^2 - 4(\omega - \omega\beta - 1)}\right]^2$$

on the domain: $\{(\alpha, \beta) | 0 \leq \alpha \leq \hat{\alpha}, \underline{\beta} \leq \beta \leq \hat{\beta}, \text{ and } \underline{\beta} \leq \alpha + \beta \leq \rho(M_J)\}$, where $\hat{\alpha} = max_{x \neq 0} x^* D(B + C)x / x^* Dx$. By calculus, we find that $\lambda(\alpha, \beta)$ has its maximum value at $\alpha = \rho(M_J) - \hat{\beta}$ and $\beta = \hat{\beta}$. Hence,

$$\rho(M_p(\omega)) \leq \lambda(\rho(M_J) - \hat{\beta}, \hat{\beta}),$$

which gives (4.12).

Further, since $0 < \rho(M_J) - \hat{\beta} < 1 - \hat{\beta}$ and $0 < \omega \leq \dfrac{2}{1 - \underline{\beta}} < \dfrac{2}{1 - \hat{\beta}}$, from Lemma 4.2 it follows $\lambda(\rho(M_J) - \hat{\beta}, \hat{\beta}) < 1$. This shows that $\rho(M_p(\omega)) < 1$ when $E \geq 0$.

When $E < 0$, using Lemma 4.3 and $0 < \omega < \dfrac{2}{1 - \underline{\beta}}$, we get

$$\rho(M_p(\omega)) = \omega(1 - \beta) - 1 \leq \omega(1 - \underline{\beta}) - 1 < 1.$$

This completes the proof of Theorem 4.2. ∎

COROLLARY 4.1 (ESTIMATION OF $\rho(M_p)$). *Let M_p be the JGS iteration matrix given in (2.8). Then*

$$\rho(M_p) \le \frac{1}{4}\left((\rho(M_J) - \hat{\beta}) + \sqrt{(\rho(M_J) - \hat{\beta})^2 + 4\hat{\beta}}\right)^2. \tag{4.14}$$

Proof. (4.14) follows from (4.12) immediately by setting $\omega = 1$. ∎

THEOREM 4.3 (APPROXIMATION OF ω_b). *Let $\varrho(\omega)$ denote the upper bound of $\rho(M_p(\omega))$ given in Theorem 4.2. Then the optimal relaxation parameter ω_b of JSOR can be approximately chosen as the minimum point ω_a of $\varrho(\omega)$ in the interval $I = (0, 2/(1-\underline{\beta}))$, where ω_a is a real root of the equation*

$$\begin{aligned}\omega^3\left(\rho(M_J) - \hat{\beta}\right)^2(1-\underline{\beta}) - \omega^2\left[(2-\hat{\beta}-\underline{\beta})^2 + \left(\rho(M_J) - \hat{\beta}\right)^2\right] \\ + 4\omega(2-\hat{\beta}-\underline{\beta}) - 4 \;=\; 0,\end{aligned} \tag{4.15}$$

and is the closest to the endpoint $2/(1-\underline{\beta})$. Further, $\varrho(\omega)$ can be written

$$\varrho(\omega) = \begin{cases} \frac{1}{4}\left[\omega(\rho(M_J) - \hat{\beta}) + \Phi\right]^2 & \text{for } 0 < \omega \le \omega_a, \\ \omega(1-\underline{\beta}) - 1 & \text{for } \omega_a \le \omega < \dfrac{2}{1-\underline{\beta}}. \end{cases} \tag{4.16}$$

where $\Phi = \sqrt{\omega^2(\rho(M_J) - \hat{\beta})^2 - 4(\omega - \omega\hat{\beta} - 1)}$.

Proof. Since $\rho(M_{SOR}(\omega)) \le \varrho(\omega)$, and ω_a satisfies $\varrho(\omega_a) = \min_{\omega \in I}\varrho(\omega)$, we can define ω_a as an approximation of ω_b.

From expressions (4.12) and (4.13) it is easy to see that $\varrho(\omega)$ is decreasing for $E \ge 0$ while increasing for $E < 0$. Hence, the minimum point ω_a of $\varrho(\omega)$ must be a root of the the following equation for ω:

$$\frac{1}{4}\left[\omega(\rho(M_J) - \hat{\beta}) + \Phi\right]^2 = \omega(1-\underline{\beta}) - 1.$$

By elementary calculations, the above equation can be simplified to (4.15). Clearly, if ω_a is a root of (4.15) in the interval I, and is the closest to the end point $2/(1-\underline{\beta})$, then $\varrho(\omega)$ has the minimum value at $\omega = \omega_a$, and can be written as (4.16). ∎

COROLLARY 4.2 (THE SOR OPTIMAL RELAXATION PARAMETER THEOREM). *Let $M_{SOR}(\omega)$ be the SOR iteration matrix. If the assumptions in Theorem 4.2 hold, then*

$$\begin{aligned}&\rho(M_{SOR}(\omega)) \\ &= \begin{cases} \frac{1}{4}\left[\omega\rho(M_J) + \sqrt{\omega^2\rho(M_J)^2 - 4(\omega-1)}\right]^2 & \text{for } 0 < \omega \le \omega_b, \\ \omega - 1 & \text{for } \omega_b \le \omega < 2, \end{cases}\end{aligned} \tag{4.17}$$

where

$$\omega_b = \frac{2}{1+\sqrt{1-\rho(M_J)^2}} \tag{4.18}$$

is the optimal relaxation parameter satisfying

$$\rho(M_{SOR}(\omega_b)) = \min_{\omega} \rho(M_{SOR}(\omega)). \tag{4.19}$$

Proof. When $p = 1$, we have $M = N = 0$, so that $\beta = 0$, implying that $\rho(M_p(\omega)) = \rho(M_{SOR}(\omega))$. Thus, by Theorem 4.2,

$$\rho(M_{SOR}(\omega)) \le \varrho(\omega), \tag{4.20}$$

and the expression (4.16) for $\varrho(\omega)$ is reduced to

$$\varrho(\omega) = \begin{cases} \frac{1}{4}\left[\omega\rho(M_J) + \sqrt{\omega^2\rho(M_J)^2 - 4(\omega-1)}\right]^2 & \text{for } 0 < \omega \le \omega_a, \\ \omega - 1 & \text{for } \omega_a \le \omega < 2. \end{cases}$$

On the other hand, it is easy to see that

$$\varrho(\omega) = \frac{1}{4}\left[\omega\rho(M_J) + \sqrt{\omega^2\rho(M_J)^2 - 4(\omega-1)}\right]^2$$

is a solution of Equation (4.6) with $\alpha = \rho(M_J)$ and $\beta = 0$. Hence, with (4.4) we can see that $\varrho(\omega)$ is an eigenvalue of $M_{SOR}(\omega)$. This gives

$$\rho(M_{SOR}(\omega)) \ge \varrho(\omega). \tag{4.21}$$

Therefore, combining (4.20) and (4.21) gives $\rho(M_{SOR}(\omega)) = \varrho(\omega)$.

Further, with $p = 1$, (4.15) becomes

$$\omega^3\rho(M_J)^2 - \omega^2\left[4 + \rho(M_J)^2\right] + 8\omega - 4 = 0,$$

which can be simplified to

$$\omega^2\rho(M_J)^2 - 4(\omega - 1) = 0.$$

Solving this equation for ω gives the expression (4.18), so that $\omega_a = \omega_b$. Therefore, (4.16) implies (4.17) and (4.19). ∎

COROLLARY 4.3 (THE DAMPED-JACOBI OPTIMAL RELAXATION PARAMETER THEOREM). *Let $M_J(\omega)$ be the damped-Jacobi iteration matrix, and μ its smallest eigenvalue. Then*

$$\rho(M_J(\omega)) = \begin{cases} 1 + \omega\rho(M_J) - \omega, & 0 < \omega \le \omega_b, \\ \omega - \omega\mu - 1, & \omega_b \le \omega < \dfrac{2}{1-\mu}, \end{cases} \tag{4.22}$$

where

$$\omega_b = \frac{2}{2-\mu-\rho(M_J)}, \tag{4.23}$$

and

$$\rho(M_J(\omega_b)) = \min_{\omega} \rho(M_J(\omega)) = \frac{\rho(M_J)-\mu}{2-\mu-\rho(M_J)}. \tag{4.24}$$

Proof. In the case of $p = n$, we have $M = U$, $N = L$ and $B = C = 0$. Thus, $\hat{\beta} = \rho(M_J)$ and $\beta = \mu$. By using Theorems 4.2 and 4.3, we have $\rho(M_J(\omega)) \leq \varrho(\omega)$, and $\varrho(\omega)$ can be expressed by (4.22).

On the other hand, (4.4) implies that $\varrho(\omega)$ must be an eigenvalue of $M_J(\omega)$, implying $\rho(M_J(\omega)) \geq \varrho(\omega)$. Hence, $\rho(M_J(\omega)) = \varrho(\omega)$.

Further, in the case of damped-Jacobi (i.e., $p = n$), Equation (4.15) becomes

$$-\Big[\omega(2-\rho(M_J)-\mu)\Big]^2 + 4\omega\Big(2-\rho(M_J)-\mu\Big) - 4 = 0,$$

which can be simplified to $[\omega(2-\rho(M_J)-\mu)-2]^2 = 0$. Solving this equation for ω gives (4.23), and then (4.24). ∎

5. A model problem analysis. We consider the following Poisson model problem

$$-\Delta u = f(x,y), \quad 0 < x,\ y < 1, \tag{5.1}$$

where $u(x,y) = g(x,y)$ on the boundary of the unit domain $(0 < x, y < 1)$.

With a given mesh size $h = \dfrac{1}{n+1}$, we obtain a second-order finite difference approximation of (5.1)

$$\begin{aligned} 4u(x,y) - u(x-h,y) - u(x+h,y) - u(x,y-h) - u(x,y+h) \\ = h^2 f(x,y), \end{aligned} \tag{5.2}$$

where $u(x,y) = g(x,y)$ when mesh point (x,y) is on the boundary.

We number mesh points in a natural ordering (i.e. from left to right and bottom to top) such that the system of equations in (5.2) is written in a form of linear system (2.1). We then partition the index set $W = \{i + (j-1)n \ | i,j = 1,2,\ldots,n\}$ into n subsets such that subset R_k for $k = 1,2,\ldots,n$ consists of the ordering numbers of the mesh points on the k-th mesh line (i.e., $R_k = \{i + (k-1)n \mid i = 1,2,\ldots,n\}$). Based on this line partition, the JGS iterates $\{u(x,y)^k\}$ for solving (5.2) are defined by

$$\begin{aligned} u(x,y)^{k+1} &= \frac{1}{4}\Big[u(x-h,y)^{k+1} + u(x+h,y)^k \\ &\quad + u(x,y-h)^k + u(x,y+h)^k + h^2 f(x,y)\Big], \end{aligned} \tag{5.3}$$

where $u(x,y)^k = g(x,y)$ for a boundary mesh point (x,y).

THEOREM 5.1. *Let ρ_{JGS} be the spectral radius of the iteration matrix of the JGS defined in (5.3). Then*

$$\rho_{JGS} = \left(\frac{\cos h\pi + \sqrt{\cos^2 h\pi + 8\cos h\pi}}{4} \right)^2. \tag{5.4}$$

Proof. Let λ and v be an eigenvalue and a corresponding eigenvector of the iteration matrix of the JGS defined in (5.3), respectively. Then, they satisfy

$$4\lambda v(x,y) = \lambda v(x-h,y) + v(x+h,y) + v(x-h,y) + v(x,y+h), \tag{5.5}$$

where $v(x,y) = 0$ when (x,y) is a boundary mesh point.

Setting $v(x,y) = \lambda^{\frac{x}{2h}} \sin \pi\mu x \sin \pi\nu y$ in (5.5), we can get

$$2\lambda = \lambda^{\frac{1}{2}} \cos \pi\mu h + \cos \pi\nu h,$$

where $\nu, \mu = 1, 2, \ldots, h^{-1} - 1$. Solving the above equation for λ gives

$$\lambda_{\mu,\nu} = \left(\frac{\cos \mu h\pi + \sqrt{\cos^2 \mu h\pi + 8\cos \nu h\pi}}{4} \right)^2.$$

Thus, the spectral radius $\rho_{JGS} = \max |\lambda_{\mu,\nu}| = \lambda_{1,1}$ as given in (5.4). ∎

Since $\cos h\pi \sim 1 - \pi^2 h^2/2$, the spectral radius ρ_{JGS} can be approximated by

$$\rho_{JGS} \sim 1 - \frac{2}{3} h^2 \pi^2.$$

From [10] we know that the Jacobi and Gauss-Seidel spectral radii ρ_J and ρ_{GS} for (5.2) have the approximate expressions

$$\rho_J \sim 1 - \frac{1}{2}\pi^2 h^2, \quad \text{and} \quad \rho_{GS} \sim 1 - \pi^2 h^2.$$

Hence, we have

$$\rho_J > \rho_{JGS} > \rho_{GS}.$$

This confirms a conclusion of Theorem 3.1. That is, JGS converges faster than Jacobi but slower than Gauss-Seidel, asymptotically.

We next consider JSOR for solving (5.2). We will show that the value ω_a given in Theorem 4.3 is a good approximation of the optimal relaxation parameter ω_b.

To compute ω_a by using Equation (4.15), we need the values of $\rho(M_J)$, $\underline{\beta}$, and $\hat{\beta}$, where $\underline{\beta}$ and $\hat{\beta}$ are the smallest and largest eigenvalues of $M+N$,

respectively. From [10] we know $\rho(M_J) = cos\pi h$. To get $\underline{\beta}$ and $\hat{\beta}$, we consider the eigenvalue problem

$$\frac{1}{4}\Big[w(x,y-h)+w(x,y+h)\Big] = \beta w(x,y)$$

(i.e., $(M+N)w = \beta w$). Setting $w(x,y) = \sin i\pi x \sin j\pi y$ for $i,j = 1,2,\ldots,h^{-1}-1$, we find $\beta = \frac{1}{2}\cos(j\pi h)$. Thus,

$$\underline{\beta} = \frac{1}{2}\cos\pi(1-h), \quad \text{and} \quad \hat{\beta} = \rho(M+N) = \frac{1}{2}\cos\pi h.$$

For simplicity, we set $h = \frac{1}{10}$. We then find the three roots of Equation (4.15) (by using *Matlab*) as follows

$$\omega_1 = 10.4892, \quad \omega_2 = 1.2929, \quad \text{and} \quad \omega_3 = 0.8840,$$

as well as $\frac{2}{1-\underline{\beta}} = 1.3554$. According to Theorem 4.3, we set $\omega_a = \omega_2 = 1.2929$. To confirm it, we plot the function $\varrho(\omega)$ in Fig. 2, showing that $\varrho(\omega)$ does have the minimum at $\omega = \omega_a = 1.2929$.

We also compute the optimal relaxation parameter ω_b directly according to the definition (4.1) by using *Matlab*. We get $\omega_b = 1.29$, and $\rho(M_p(\omega_b)) \approx 0.9079$.

We then numerically test the performance of JSOR using $\omega_b = 1.29$ and $\omega_a = 1.2929$ for solving (5.2) with $f(x,y) = g(x,y) = 0$ and $h = \frac{1}{10}$. We use the initial guess $u^0 = 1$, and the convergence criterion

$$\frac{\|f - Au^j\|_2}{\|f\|_2} \le \epsilon \tag{5.6}$$

with $\epsilon = 10^{-6}$. We find that JSOR using $\omega = \omega_a$ takes only one more iteration than using $\omega = \omega_b$ to satisfy (5.6) (one is 174 and the other 173). This shows that the value ω_a given by Theorem 4.3 is a good approximation to the optimal relaxation parameter ω_b of JSOR.

We also test JSOR using other values of ω, and plot the total number of JSOR iterations as a function of ω in Fig. 3. It confirms that ω_b is the optimal relaxation parameter, and JSOR is divergent when $\omega \ge \frac{2}{1-\underline{\beta}} = 1.3554$.

6. Parallel performance of JSOR. In this section, we demonstrate the parallel performance of JSOR by considering the model problem (5.2) with $f(x,y) = 2\pi^2 \sin\pi x \sin\pi y$ and $h = 1/161$. For simplicity, the grid mesh is partitioned into p strips to implement on p processors for $p =$

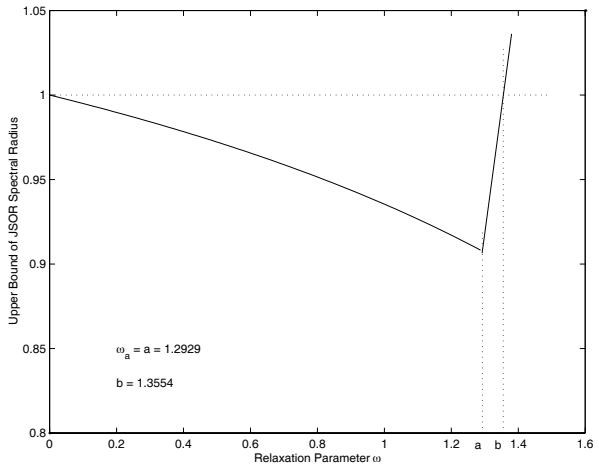

FIG. 2. *The upper-bound function $\varrho(\omega)$, defined in (4.16), of the spectral radius $\rho(M_p(\omega))$ of JSOR for solving (5.2) with $h = \frac{1}{10}$. Here b denotes $2/(1-\underline{\beta})$. It shows that $\varrho(\omega)$ has the minimum value at ω_a, and $\varrho(\omega) \geq 1$ for $\omega \geq \frac{2}{1-\underline{\beta}}$.*

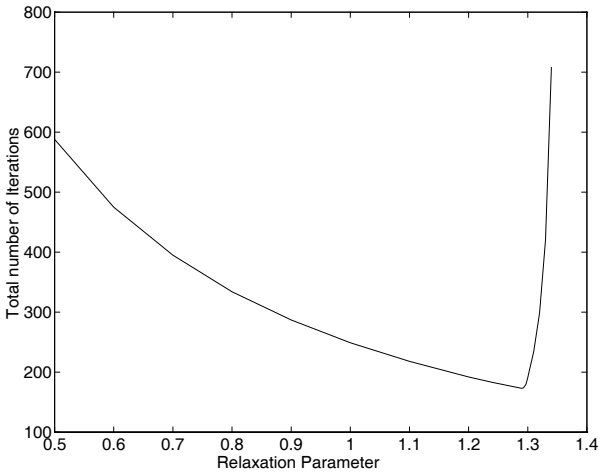

FIG. 3. *The total number of JSOR iterations for solving (5.2) with $f(x,y) = g(x,y) = 0$ and $h = \frac{1}{10}$ as a function of relaxation parameter ω. It has the minimum value (173) at $\omega = \omega_b = 1.29$. The figure also shows that JSOR is divergent for $\omega \geq \frac{2}{1-\underline{\beta}} = 1.3554$.*

1, 2, 4, 8, and 16. An initial guess of zero is taken. The iterative convergence criterion is (5.6) with $\epsilon = 10^{-4}$.

As a comparison of JSOR, we also implemented Jacobi on p processors. Numerical experiments were made on a shared memory MIMD parallel computer, the KSR1 at the University of Houston. The parallel programs for JSOR and Jacobi were compiled using optimization level $-O2$. CPU times were computed by using the system function, *user_seconds*(). The optimal relaxation parameters of JSOR and SOR were estimated by numerical experiments.

We first recall some standard terminology regarding parallel computation: *Speedup* is $S(p) = T(1)/T(p)$, where $T(p)$ is the CPU time needed for solving the given problem on p processors [4], and *Linear speedup* refers to the ideal case when $T(p)$ equals $T(1)/p$. *Total Time* is the CPU time spent from the beginning of the iteration until the convergence criterion (5.6) is satisfied (here it does not include the CPU time spent on the calculation of f, the initial guess, and the input/output of data). *Comm. Time* is the CPU time spent on the interprocessor-data communication. *Comp. Time* is the CPU time only spent on the computation of the iteration and the L_2-norm of the residual.

Fig. 4 displays the parallel performance of JSOR for solving (5.2) with $h = 1/161$. Here JSOR used the optimal relaxation parameters for each value of p. From the figure we see that *Comm. Time* is very small, indicating that JSOR can be efficiently implemented on a parallel computer. But, from the figure we also see that JSOR takes almost the same *Total Time* for each group of processors since the total number of JSOR iterations increased almost linearly with respect to p as shown in Fig. 7. Hence, JSOR does not provide a suitable parallel version of SOR since no *Speedup* is achieved.

Fig. 5, however, shows that JGS, a particular case of JSOR with $\omega = 1$, can take significantly less *Total Time* by using more processors. In fact, from Fig. 7 we see that the total number of JGS iterations is a slowly increasing function of p. Hence, compared to the sequential Gauss-Seidel, JGS can have substantial *Speedup*. For this example, JGS has $S(p) = 9$ for $p = 16$.

As a comparison, we also implemented Jacobi for solving the same problem and based on the same domain partitioning as JSOR. We get the speedup $S(p) = 12$ for Jacobi for $p = 16$. Even though Jacobi has a larger *Speedup* than JGS, Jacobi takes much more *Total Time* due to its slower convergence speed than JGS (see Figs. 6 and 7). Hence, JGS has significantly improved the parallel performance of Jacobi. Moreover, the convergence speed of JGS is improved further by JSOR using the optimal relaxation parameter as shown in Fig. 7. Indeed, even with one processor JSOR (or SOR) using the optimal relaxation parameter can take much less CPU time than the JGS using many processors (see Fig. 6).

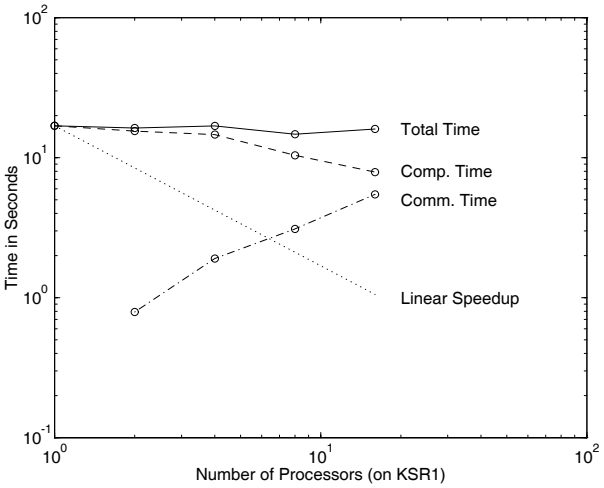

FIG. 4. *The parallel performance of the JSOR using* $\omega = \omega_b(p)$.

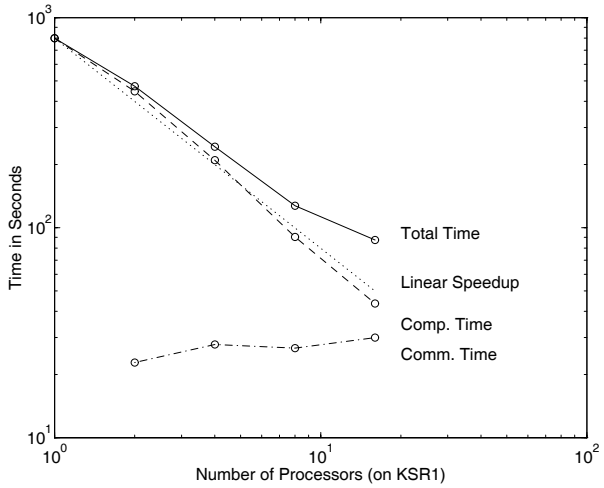

FIG. 5. *The parallel performance of JGS (a particular case of JSOR with* $\omega = 1$*).*

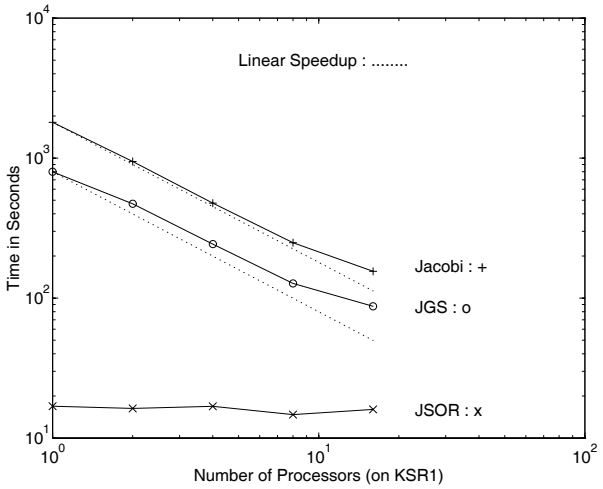

FIG. 6. *Dependence of* Total Time *on the number of processors.*

Obviously, the convergence rate of JSOR can be improved by applying more than one step of SOR to the solution of each block equation in (2.4). This is a common way to define an "incomplete" block Jacobi scheme. But, we observe that even for the two-block Jacobi method (i.e., set $p = 2$, and solve each block equation in (2.4) exactly), the spectral radius may be much larger than the spectral radius of SOR using the optimal relaxation parameter. For example, we find that the spectral radii of the two-block Jacobi method and the SOR method using $\omega_b = 1.491$ for solving (5.1) with $h = 1/9$ are 0.6848 and 0.491, respectively. Hence, the two-block Jacobi has a much slower convergence speed than the SOR using the optimal relaxation parameter. So, *it is not possible to generate a scheme from the "incomplete" block Jacobi approach that can be competitive with SOR using the optimal relaxation parameter.*

Finally, we made numerical experiments using more than one step of SOR to solve each block equation in (2.4) for the same model problem as in Fig. 4 on 16 processors of the KSR1. Numerical results are listed in Table 1. They show that *Total Time* is an increasing function of the number of SOR iterations used in solving each block equation in (2.4) even though the total number of the resulting "incomplete" block Jacobi iterations is a decreasing function. This suggests that the JSOR method (i.e., apply only one SOR to each block equation in (2.4)) can be the best scheme we can define from the "incomplete" block Jacobi approach.

7. Conclusions. We have analyzed a parallel linear stationary iterative method, the JSOR method, theoretically and numerically. Defined by using a domain partition technique, JSOR can be easily and efficiently im-

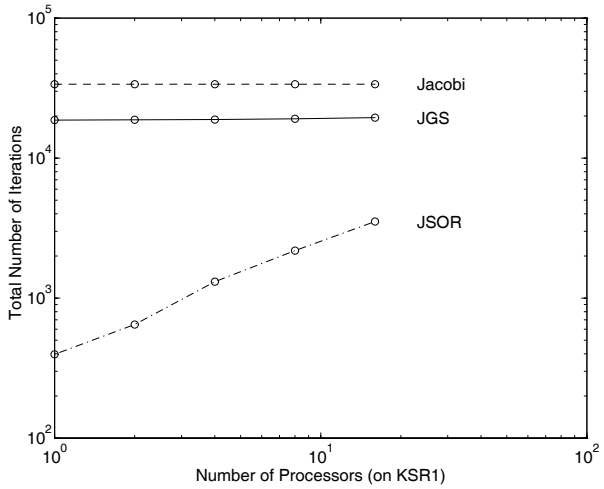

FIG. 7. *Dependence of the total number of iterations on the number of processors.*

TABLE 1
Dependence of the performance of "incomplete" block Jacobi on the number of SOR iterations applied to each block equation in (2.4) for solving the model problem (5.2) with $h = 1/161$ *and* $p = 16$.

SOR Itns. in each block	1	2	4	8	16
Block Jacobi Itns.	2845	2220	2078	1966	1906
Total Time in seconds	12.48	14.09	19.95	31.67	55.06
ω_b for SOR	1.80	1.84	1.85	1.86	1.80

plemented on a MIMD parallel computer, especially for solving complicated scientific problems (to which it may be difficult to apply the Red/Black SOR method, a widely-used parallel version of SOR). JSOR has different parallel iterative expressions based on different domain partitions. In particular, JSOR contains the SOR and damped-Jacobi methods as two extreme cases. In this sense, the JSOR analysis can provide a general linear stationary iterative theory, including the classic SOR and damped-Jacobi theories as special cases.

We have proved several basic convergence theorems for JSOR, including one concerning the optimal relaxation parameter. We also have proved a comparison theorem for the convergence rates of JGS, Jacobi and Gauss-Seidel, and shown that the SOR and damped-Jacobi theorems follow directly from the corresponding JSOR theorems. Further, we have presented a model problem analysis for JSOR to confirm our theoretical results.

Numerical results on a parallel computer have been presented to demonstrate the parallel performance of JSOR. They have shown that

JSOR can communicate interprocessor data as efficiently as Jacobi, and can significantly improve the convergence rate of Jacobi. For JGS (i.e., JSOR with $\omega = 1$), it has been shown to have a satisfactory speedup compared to the Gauss-Seidel method. But, in the case of optimal relaxation parameter, we have observed that the convergence speed of JSOR might be reduced almost linearly with respect to the partition number p, so that it does not provide a suitable parallel version of the sequential SOR method since it has no speedup. Consequently, JSOR is not a satisfactory solver of linear systems. Instead, a principal application of JSOR is as a parallel smoother in the parallel multigrid method. Based on the JSOR analysis given here, we have proposed a JSOR smoothing analysis in [9], showing that a JSOR smoother can be more robust than the damped-Jacobi smoother, and can be as robust as an SOR smoother when p is not too large.

Numerical results in this paper also have shown that even the two-block Jacobi method may have a much slower convergence speed than SOR using the optimal relaxation parameter. This suggests that we cannot expect a parallel linear stationary iteration defined only by using an "incomplete" block Jacobi approach to have the same convergence rate as SOR. This observation motivated the development of the PSOR method in [7] to obtain a parallel linear stationary iterative method by domain partitioning with the same convergence rate as SOR and the same advantages as JSOR in the parallel implementation.

REFERENCES

[1] B. Bagheri, A. Ilin, and L.R. Scott, *Parallel 3-D MOSFET simulation*, Proceedings of the 27th annual Hawaii international conference on system sciences, **1**:46–54, 1994.

[2] W. Hackbusch, *Multigrid Methods and Applications,* Springer-Verlag, New York, 1985.

[3] D.P. Bertsekas and J.N. Tsitsiklis, *Parallel and Distributed Computation*, Prentice Hall, New Jersey, 1989.

[4] M.J. Quinn, *Designing Efficient Algorithms for Parallel Computers*, McGraw-Hill, Inc., 1987.

[5] B. Smith, P. Bjorstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 1996.

[6] R. Varga, *Matrix Iterative Analysis*, Prentice Hall, Englewood Cliffs, New Jersey, 1962.

[7] D. Xie and L. Adams, *New parallel SOR method by domain partitioning*, SIAM J. Sci. Comput., **20**(6):2261–2281, 1999.

[8] D. Xie and L.R. Scott, *The parallel U-cycle multigrid method*, UH/MD Technical Report, **240**, University of Houston, 1997.

[9] D. Xie, *New Parallel Iteration Methods, New Nonlinear Multigrid Analysis, and Application in Computational Chemistry*, Research Report UH/MD, **208**, University of Houston, 1995.

[10] D.M. Young, *Iterative Solution of Large Linear System*, Press Academic, New York, 1971.