THE SIGMA-SOR ALGORITHM AND THE OPTIMAL STRATEGY FOR THE UTILIZATION OF THE SOR ITERATIVE METHOD

ZBIGNIEW I. WOŹNICKI

ABSTRACT. The paper describes, discusses, and numerically illustrates the method for obtaining a priori estimates of the optimum relaxation factor in the SOR iteration method. The computational strategy of this method uses the so-called Sigma-SOR algorithm based on the theoretical result proven in the paper. The method presented is especially efficient for problems with slowly convergent iteration process and in this case is strongly competitive with adaptive procedures used for determining dynamically the optimum relaxation factor during the course of the SOR solution.

1. Introduction

The SOR (Successive Over-Relaxation) method and its line variants are among the most popular and efficient iterative methods used for solving large and sparse linear systems of equations arising in many areas of science and engineering. The popularity of SOR algorithms is in a great measure due to their simplicity from the programming point of view. The rate of convergence of the SOR method depends strongly on the relaxation factor \( \omega \); therefore, the main difficulty in the efficient use of this method lies in making a good estimate of the optimum relaxation factor \( \omega_{\text{opt}} \) which maximizes the rate of convergence.

For a large class of matrix problems arising in the discretization of elliptic partial differential equations the coefficient matrices have certain eigenvalue properties allowing us to determine explicitly the optimum relaxation factor \( \omega_{\text{opt}} \). In the case when the coefficient matrix is 2-cyclic and consistently ordered \([1]\) (this property will be assumed in the remainder), \( \omega_{\text{opt}} \) can be determined by finding the value of the spectral radius \( \rho(\mathcal{L}_1) \) for the associated Gauss-Seidel iteration matrix \( \mathcal{L}_1 \).

However, it is well known that the nature of the dependence of \( \omega_{\text{opt}} \) on \( \rho(\mathcal{L}_1) \) indicates the sensitivity of the rate of convergence to the accuracy in determining \( \omega_{\text{opt}} \), as \( \rho(\mathcal{L}_1) \) approaches unity \([1, 2]\). When \( \rho(\mathcal{L}_1) \) is very close to unity, small changes in the estimate of \( \rho(\mathcal{L}_1) \) can seriously decrease the rate of convergence, and just in this case the availability of an accurate value of \( \rho(\mathcal{L}_1) \) is an essential point for the efficient use of the SOR method.

Received by the editor October 9, 1992.

1991 Mathematics Subject Classification. Primary 65B99, 65F10, 65F15, 65F50.

Key words and phrases. SOR iteration method, power method, acceleration of convergence, eigenvalues of iteration matrix, estimation of optimum relaxation factor.
In practice two approaches are used to determine $\omega_{\text{opt}}$. One approach proposed in the literature [2, 3, 4] is determining $\omega_{\text{opt}}$ dynamically, as the SOR iteration proceeds with using some $\omega_i < \omega_{\text{opt}}$. Then by examining certain conditions for quantities derived from current numerical results, $\omega_i$ is updated to a new relaxation factor $\omega_{i+1} \leq \omega_{\text{opt}}$ until the assumed tolerance criterion is satisfied.

The second approach for determining $\omega_{\text{opt}}$ is based on obtaining an a priori estimation of $\rho(L)$, usually by means of the power method or its modifications. As is well known, the rate of convergence of the power method is governed by the ratio of the largest subdominant (in the absolute value) to the dominant eigenvalue. If this ratio is close to unity, the power method will converge very slowly and in such a case determining $\omega_{\text{opt}}$ may be more time-consuming than the SOR iteration itself.

Basically, there is no general comparison procedure to determine which approach is "best". However in the case of 2-cyclic consistently ordered matrices, an accurate estimate for $\rho(L)$ prior to the SOR iteration solution can be effectively obtained by an appropriate use of power method iterations, and this topic is the main purpose of the paper.

In the next section the SOR iterative method and the power method are briefly described, and well-known basic results are recalled. These basic results are essential in deriving the Sigma-SOR algorithm. The computational strategy for determining the optimum relaxation factor $\omega_{\text{opt}}$ is described in the third subsection of §2.

The secondary purpose of this paper, discussed in §3, is to give numerical results for a variety of problems presented in the literature in order to illustrate the efficiency of the proposed method for the a priori determination of the optimum relaxation factor $\omega_{\text{opt}}$.

2. Formulation

2.1. The SOR iteration method. In the iterative solution of the linear system

$$Ax = b$$

the $n \times n$ matrix $A$ is usually defined by the following decomposition:

$$A = D - L - U,$$

where $D$, $L$, and $U$ are diagonal, strictly lower triangular and strictly upper triangular matrices, respectively.

The SOR iterative method [1] is defined by

$$Dx^{(t+1)} = \omega[Lx^{(t+1)} + Ux^{(t)} + b] - (\omega - 1)Dx^{(t)}, \quad t = 0, 1, 2, \ldots$$

or equivalently, if $D$ is a nonsingular matrix

$$x^{(t+1)} = \mathcal{L}\omega x^{(t)} + (D - \omega L)^{-1}b,$$

where

$$\mathcal{L}\omega = (D - \omega L)^{-1}[\omega U - (\omega - 1)D]$$

is called the SOR iteration matrix and $\omega$ is the relaxation factor. For $\omega = 1$ the SOR method reduces to the classical scheme known as the Gauss-Seidel iterative method and

$$\mathcal{L}_1 = (D - L)^{-1}U$$

is called the Gauss-Seidel iteration matrix.
In the point algorithm, the iteration proceeds for one component of the approximate solution vector at a time. For block or line algorithms, the iteration involves improving simultaneously groups of components, and therefore they are more efficient than the point algorithm. In this case the matrices $D$, $L$, and $U$ have a block structure corresponding to the assumed partitioning of components.

It is well known [1] that in the case of 2-cyclic consistent orderings, when the associated nonnegative Jacobi iteration matrix

$$B = D^{-1}(L + U) \geq 0$$

is convergent (i.e., $\rho(B) < 1$), then $\mathcal{L}_1$ has only nonnegative eigenvalues $\lambda_i$ such that

$$1 > \rho(\mathcal{L}_1) = \lambda_1 > \lambda_2 > \lambda_3 > \cdots,$$

and the following fundamental relation due to Young (see, for example [1] and the references given therein) holds between $\lambda_i$ and the corresponding eigenvalues $\nu_i$ of $\mathcal{L}_\omega$:

$$\lambda_i = \frac{1}{\nu_i} \left[ \nu_i + \omega - 1 \right]^2.$$

Moreover, $\rho(\mathcal{L}_\omega) = \max_{1 \leq i \leq n} |\nu_i| < 1$ for $0 < \omega < 2$, and its minimum value is attained when

$$\omega = \omega_{opt} = \bar{\omega} = \frac{2}{1 + \sqrt{1 - \rho(\mathcal{L}_1)}},$$

in which case

$$\rho(\mathcal{L}_\omega) = \bar{\omega} - 1 = \frac{1 - \sqrt{1 - \rho(\mathcal{L}_1)}}{1 + \sqrt{1 - \rho(\mathcal{L}_1)}}.$$

In the convergence analysis of iterative methods the (asymptotic) rate of convergence

$$R(\mathcal{G}) = -\ln \rho(\mathcal{G})$$

is certainly the simplest practical measure of the speed of convergence for a convergent matrix $\mathcal{G}$. The rate of convergence is especially useful for comparing the efficiency of different iterative methods, because the number of iterations $t$ required for reducing the error norm in a given method by a prescribed factor $f$ is roughly inversely proportional to the rate of convergence, and is given by

$$t \approx \frac{-\ln f}{R(\mathcal{G})},$$

where $\mathcal{G}$ is the iteration matrix of the method.

Thus, the efficiency of different iterative methods (with a similar arithmetical effort per iteration) can be theoretically evaluated by a comparison of the rate of convergence. The data given in Table 1 (next page) illustrate the efficiency of the SOR method by comparing it with the Gauss-Seidel method, where

$$E_t = \frac{R(\mathcal{L}_\omega)}{R(\mathcal{L})}$$

is the theoretical coefficient of efficiency and $\bar{\omega} \equiv \omega_{opt}$. 

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use
As can be seen from Table 1, the efficiency of the SOR method drastically increases as $\rho(\mathcal{L}_1)$ becomes close to unity. For the case when $\rho(\mathcal{L}_1) = 0.9999$, the SOR method is asymptotically 200 times faster than the Gauss-Seidel method. Since $\omega_{\text{opt}}$ is a function only of the spectral radius $\rho(\mathcal{L}_1)$, then for any efficient use of the SOR method, computing an accurate value of $\rho(\mathcal{L}_1)$ is needed, and the order of the accuracy in estimating $\rho(\mathcal{L}_1)$ is dependent on the closeness of $\rho(\mathcal{L}_1)$ to unity.

2.2. The power method. Usually, an estimate for $\rho(\mathcal{L}_1)$ is obtained by using the ordinary power method [5], which will be used in the analysis presented in this paper. The power method is conceptually and computationally the simplest iterative procedure for approximating the eigenvector corresponding to the dominant (largest in modulus) eigenvalue of a given matrix $\mathcal{G}$. It is defined by the iterative process

$$z^{(t)} = \mathcal{G}z^{(t-1)} = \mathcal{G}^2z^{(t-2)} = \cdots = \mathcal{G}^t z^{(0)}, \quad t = 1, 2, \ldots,$$

which converges for almost any randomly chosen nonzero starting vector $z^{(0)}$.

We assume, throughout this paper, that the $n \times n$ real matrix $\mathcal{G}$ has $n$ linearly independent eigenvectors $u_i$, and its eigenvalues $\lambda_i$ will be ordered such that

$$\lambda_1 > |\lambda_2| > |\lambda_3| > \cdots \geq |\lambda_n|.$$  

Since by assumption, $\mathcal{G}$ has a complete set of eigenvectors $u_i$, an arbitrary nonzero vector $z^{(0)}$ can be expressed in the form

$$z^{(0)} = \sum_{i=1}^{n} a_i u_i,$$

where $a_i$ are scalars not all zero.

Then the sequence (14) has the representation

$$z^{(t)} = \sum_{i=1}^{n} a_i \lambda_i^t u_i = \lambda_1^t \left[ a_1 u_1 + \sum_{i=2}^{n} a_i (\lambda_i / \lambda_1)^t u_i \right] = \lambda_1^t [a_1 u_1 + e^{(t)}].$$

Since $|\lambda_i / \lambda_1| < 1$ for all $i \geq 2$, it is clear that $z^{(t)}$ converges to $u_1$ as $t \to \infty$, provided only that $a_1 \neq 0$.

Thus the vector $z^{(t)}$ is an approximation to an unnormalized eigenvector $u_1$ belonging to $\lambda_1$, which can be considered as accurate if $\|e^{(t)}\|$ is sufficiently small. Since

$$z^{(t+1)} = \lambda_1^{t+1} [a_1 u_1 + e^{(t+1)}],$$

As can be seen from Table 1, the efficiency of the SOR method drastically increases as $\rho(\mathcal{L}_1)$ becomes close to unity. For the case when $\rho(\mathcal{L}_1) = 0.9999$, the SOR method is asymptotically 200 times faster than the Gauss-Seidel method. Since $\omega_{\text{opt}}$ is a function only of the spectral radius $\rho(\mathcal{L}_1)$, then for any efficient use of the SOR method, computing an accurate value of $\rho(\mathcal{L}_1)$ is needed, and the order of the accuracy in estimating $\rho(\mathcal{L}_1)$ is dependent on the closeness of $\rho(\mathcal{L}_1)$ to unity.

### Table 1

<table>
<thead>
<tr>
<th>$\rho(\mathcal{L}_1)$</th>
<th>$\rho(\mathcal{L}_\infty)$</th>
<th>$E_t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9</td>
<td>0.5195</td>
<td>6</td>
</tr>
<tr>
<td>0.99</td>
<td>0.8182</td>
<td>20</td>
</tr>
<tr>
<td>0.999</td>
<td>0.9387</td>
<td>63</td>
</tr>
<tr>
<td>0.9999</td>
<td>0.9802</td>
<td>200</td>
</tr>
</tbody>
</table>
it follows that for any jth component \( z_j \) of the vector \( z \),

\[
\frac{z_j^{(t+1)}}{z_j^{(t)}} = \lambda_1 \left[ \frac{a_1(u_1)_j + (e^{(t+1)})_j}{a_1(u_1)_j + (e^{(t)})_j} \right] \rightarrow \lambda_1 \quad \text{as} \ t \rightarrow \infty.
\]

The above result leads to computing the dominant eigenvalue by means of successive approximations of the corresponding eigenvector in the simple power method.

In practice, in order to keep the components of \( z^{(t)} \) within the range of practical calculation, its components are scaled at each iteration step, and (14) is replaced by the pair of equations

\[
y^{(t)} = \mathcal{F} z^{(t-1)},
\]

\[
z^{(t)} = \frac{y^{(t)}}{\|y^{(t)}\|_p},
\]

and in this case,

\[
z^{(t)} \rightarrow u_1 / \|u_1\|_p
\]

and

\[
\|y^{(t)}\|_p \rightarrow \lambda_1 \quad \text{as} \ t \rightarrow \infty,
\]

where two norms, either the maximum norm \( \| \cdot \|_\infty \) or the Euclidean norm \( \| \cdot \|_2 \), are most commonly used.

The rate of convergence will depend on the constants \( a_i \), but more essentially on the separation of the dominant eigenvalue from the largest subdominant eigenvalues of \( \mathcal{F} \), that is, on the ratios \( |\lambda_2|/\lambda_1, |\lambda_3|/\lambda_1, \ldots, \) and it is evident that the smaller these values, the faster the convergence. However, it may occur that if \( z^{(0)} \) is chosen as almost orthogonal to \( u_1 \), then \( a_1 \) in (17) will be quite small compared to the other coefficients, and whence for appropriate "small" values of \( t \), \( |a_1 \lambda_1^t| \ll |a_2 \lambda_2^t| \) and the ratio \( z_j^{(t+1)}/z_j^{(t)} \) will better approximate \( \lambda_2 \) than \( \lambda_1 \), assuming of course that \( \lambda_1 > |\lambda_2| \). In the case when \( a_1 = 0 \), the power method converges theoretically to the second eigenvector. However, in practice rounding errors will introduce small components \( u_1 \) into the vector \( z^{(t)} \) and those components will be magnified in subsequent iterations. Whence, convergence is still likely to be to the first eigenvector, although with a larger number of iterations than in the case when a more suitable starting vector \( z^{(0)} \) would be chosen.

In particular, if \( |\lambda_2|/\lambda_1 \) is close to unity, the accuracy of \( z^{(t)} \) will be proportional to \( (|\lambda_2|/\lambda_1)^t \) and the convergence may be intolerably slow, but still to the dominant eigenvalue \( \lambda_1 \). In such cases some practical techniques such as a shift of origin, or Aitken's \( \delta^2 \)-process [5], can be used to speed up the convergence of the simple power method.

In general, when \( \lambda_1 \) is the principal eigenvalue, the ratio

\[
\sigma = \max_i \frac{\lambda_i}{\lambda_1}, \quad 2 \leq i \leq n
\]

will be called the subdominance ratio, which with the assumed ordering of \( \lambda_i \) according to (15) is equivalent to

\[
\sigma = |\lambda_2|/\lambda_1.
\]
However, it seems that from the terminology point of view some comments are necessary. In the literature for the term “dominance ratio” is usually used by some authors. But it is also interesting to notice that other authors (especially the authors of books dealing with the convergence analysis of eigenvalue problems) do not use the term “dominance ratio” at all. In the author’s feeling the term “subdominance ratio” for \( \sigma \) seems to be more appropriate because \( \sigma \) increases with the absolute value of the largest subdominant eigenvalue, and the dominance of the principal eigenvalue decreases.

Since the convergence to the dominant eigenvalue by the power method is geometric in the subdominance ratio \( \sigma \), then by an analogy to the analysis of iterative methods for solving linear systems of equations one can define the (asymptotic) rate of convergence as

\[
R(\mathcal{F}) = -\ln \sigma,
\]

which is a useful measure for the speed of convergence to the dominant eigenvalue of a given matrix \( \mathcal{F} \) in the power method.

Referring back to the SOR method, we find it convenient to first consider the behavior of the eigenvalues \( \nu_i \) of \( \mathcal{L}_0 \) as a function of \( \omega \) for the case of 2-cyclic consistently ordered nonsingular matrices \( A = D - L - U \) of (1), where \( D \), \( L \), and \( U \) are nonsingular diagonal, strictly lower triangular and strictly upper triangular nonnegative matrices, respectively. As is well known \([1]\), the eigenvalues \( \nu_i \) of \( \mathcal{L}_0 \) are related by (8) to the eigenvalues \( \lambda_i \) of the Gauss-Seidel iteration matrix \( \mathcal{L}_1 \), the special case of \( \mathcal{L}_0 \) with \( \omega = 1 \). The matrix \( \mathcal{L}_1 \) has at least half the eigenvalues equal to zero, and the remaining ones are positive and real, and such that

\[
1 > \rho(\mathcal{L}_1) = \lambda_1 > \lambda_2 > \lambda_3 > \cdots.
\]

In the analysis of convergence properties of the SOR method, it is very useful to investigate the behavior of the roots of (8),

\[
\nu_i^+ , \nu_i^- = \frac{1}{2} \left[ \omega^2 \lambda_i \pm \sqrt{\omega^2 \lambda_i^2 - 4(\omega - 1)} \right] - (\omega - 1).
\]

Thus, when \( \omega = 1 \), it is clear that \( \nu_i^+ = \lambda_i \) and \( \nu_i^- = 0 \). As \( \omega \) increases from unity, \( \nu_i^+ \) and \( \nu_i^- \) are decreasing and increasing functions of \( \omega \), respectively, until \( \omega^2 \lambda_i - 4(\omega - 1) = 0 \), which occurs when

\[
\omega = \bar{\omega}_i = \frac{2}{1 + \sqrt{1 - \lambda_i}}
\]

and both roots coincide with the same value, that is, \( \nu_i^+ = \nu_i^- = \bar{\omega}_i - 1 \). For \( \omega > \bar{\omega}_i \), the roots \( \nu_i^+ \) and \( \nu_i^- \) become complex conjugate pairs and increase, the absolute value being \( \omega - 1 \). It is obvious that, for

\[
1 \leq \omega \leq \bar{\omega}_1 = \frac{2}{1 + \sqrt{1 - \lambda_1}},
\]

\( \rho(\mathcal{L}_0) = \nu_i^+ \) is a real and strictly decreasing function of \( \omega \) while for \( \bar{\omega}_1 < \omega \leq 2 \) one has \( \rho(\mathcal{L}_0) = |\omega - 1| \).

However, we should add a note about negative eigenvalues \( \nu_i \) which may exist. The matrix \( \mathcal{L}_1 \) has \( s \) (usually half of \( n \)) eigenvalues positive and \( n - s \) zero. These positive eigenvalues \( \lambda_i \) give rise to the roots \( \nu_i^+ \) while the zero
eigenvalues \( \lambda_{i+s} \) give rise to the roots \( \nu_j^- \) with \( \omega = 1 \). If \( 2s < n \), then there are zero eigenvalues \( \lambda_j \), where \( 2s < j \leq n \), which satisfy also the relation (8); hence the corresponding eigenvalues \( \nu_j = -(\omega - 1) \) are negative for all \( \omega > 1 \).

The typical behavior of the eigenvalues \( \nu_i \) of \( \mathcal{L}_\omega \) versus \( \omega \) is shown in Figure 1 for the example in which the three largest eigenvalues of \( \mathcal{L}_1 \) are \( \lambda_1 = 0.98 \), \( \lambda_2 = 0.94 \), and \( \lambda_3 = 0.9 \), and the subdominance ratio \( \sigma_1 = \lambda_2/\lambda_1 = 0.9592 \). As can be seen from Figure 1, there exist only two positive eigenvalues \( \nu_1^+ \) and \( \nu_2^+ \) for \( \bar{\omega}_3 < \omega \leq \bar{\omega}_2 \), only one \( \nu_1^+ \) for \( \bar{\omega}_2 < \omega \leq \bar{\omega}_1 \), and for \( \omega > \bar{\omega}_1 \) all eigenvalues \( \nu_i \) are complex (and negative if they exist) with the absolute value equal to \( \omega - 1 \).

It is obvious that the subdominance ratio \( \sigma_\omega \) for the SOR matrix \( \mathcal{L}_\omega \) is a function of \( \omega \) and \( \sigma_\omega = \sigma_1 = \lambda_2/\lambda_1 \) when \( \omega = 1 \). For \( 1 < \omega \leq \bar{\omega}_2 \), \( \sigma_\omega = \nu_2/\nu_1 \) is a strictly decreasing function as \( \omega \) increases from unity (because \( \nu_1 \equiv \nu_1^+ \) decreases much less rapidly than \( \nu_2 \equiv \nu_2^+ \)) and at

\[
\omega = \bar{\omega}_2 = \frac{2}{1 + \sqrt{1 - \lambda_2}}
\]

achieves its minimum \( \bar{\sigma}_\omega = \bar{\nu}_2/\nu_1 = (\bar{\omega}_2 - 1)/\nu_1 \). For \( \bar{\omega}_2 < \omega \leq \bar{\omega}_1 \), \( \sigma_\omega = |\omega - 1|/\nu_1 \) is a strictly increasing function of \( \omega \) and for all \( \bar{\omega}_1 \leq \omega \leq 2 \), \( \sigma_\omega = 1 \) because all eigenvalues \( \nu_i \) have the same absolute value equal to \( |\omega - 1| \).

In the example shown in Figure 1, the dashed curve illustrates the behavior of \( \sigma_\omega \) versus \( \omega \), where the minimum \( \bar{\sigma}_\omega = 0.6639 \) occurs at \( \bar{\omega}_2 = 1.6065 \). In terms of the rate of convergence the theoretical coefficient of efficiency

\[
\mathcal{E}_t = \frac{\mathcal{R}(\sigma_\omega)}{\mathcal{R}(\sigma_1)}
\]

is equal to 9.84. Thus for this example the computation of \( \rho(\mathcal{L}_\omega) \) by means of the power method with \( \omega = \bar{\omega}_2 \) is asymptotically about 10 times faster than with \( \omega = 1 \).
2.3. The Sigma-SOR algorithm and computational strategy. The observations in the previous subsection show the existence of the minimum value $\sigma_\omega < \sigma_1$ and moreover they allow us to precisely identify its locality which occurs at $\omega = \overline{\omega}_2$ minimizing the value of the subdominant eigenvalue $\nu_2$. The question now arises whether there exists a mathematical basis for determining the value of $\sigma_\omega$ in dependence on $\sigma_1 = \lambda_2/\lambda_1$. The following theorem gives an answer to this question.

**Theorem.** Let $\nu_i$ be the eigenvalues of the $n \times n$ SOR iteration matrix

$$\mathcal{L}_\omega = (D - \omega L)^{-1}[\omega U - (\omega - 1)D]$$

and let $\lambda_i$ be the eigenvalues of the Gauss-Seidel iteration matrix

$$\mathcal{L}_1 = (D - L)^{-1}U.$$ 

If the eigenvalues of both matrices are related by

$$\nu_i = \frac{1}{\nu_i} \left[ \frac{\nu_i + \omega - 1}{\omega} \right]^2$$

and $\mathcal{L}_1$ has only nonnegative real eigenvalues such that

$$1 > \lambda_1 > \lambda_2 > \lambda_3 > \ldots,$$

then the subdominance ratio $\sigma_\omega = \nu_2/\nu_1$ of $\mathcal{L}_\omega$ achieves its minimum $\overline{\sigma}_\omega = \overline{\nu}_2/\nu_1$ with

$$\omega = \overline{\omega}_2 = \frac{2}{1 + \sqrt{1 - \lambda_2}} = \frac{2}{1 + \sqrt{1 - \sigma_1 \lambda_1}}$$

and is defined by the following formula:

$$\overline{\sigma}_\omega = \frac{2}{1 + \sqrt{1 - \sigma_1}} - 1 = \frac{1 - \sqrt{1 - \sigma_1}}{1 + \sqrt{1 - \sigma_1}},$$

where $\sigma_1 = \lambda_2/\lambda_1$.

**Proof.** By using (29), one obtains that

$$\sigma_1 = \frac{\lambda_2}{\lambda_1} = \frac{\nu_1}{\nu_2} \left[ \frac{\nu_2 + \omega - 1}{\nu_1 + \omega - 1} \right]^2 = \sigma_\omega \left[ \frac{1 + \nu_2 - 1}{1 + \sigma_\omega \frac{\nu_2 - 1}{\nu_2}} \right]^2$$

or equivalently

$$\sigma_1 = \frac{\nu_1}{\nu_2} \left[ \frac{\sigma_\omega + \nu_2 - 1}{\sigma_\omega + \nu_2 - 1} \right]^2.$$

The proof follows immediately from a close inspection of (33). As was already stated, $\sigma_\omega$ is minimized when $\omega = \overline{\omega}_2$ and its value is $\sigma_\omega = \overline{\sigma}_\omega = \overline{\nu}_2/\nu_1$, where $\overline{\nu}_2 = \overline{\omega}_2 - 1$. Hence, for $\omega = \overline{\omega}_2$, (33) reduces to

$$\sigma_1 = \sigma_\omega \left[ \frac{2}{1 + \sigma_\omega} \right]^2$$

and has the solution

$$\overline{\sigma}_\omega = \frac{2}{1 + \sqrt{1 - \sigma_1}} - 1 = \frac{1 - \sqrt{1 - \sigma_1}}{1 + \sqrt{1 - \sigma_1}}.$$

This completes the proof of the theorem. □
It is necessary, however, to make some comments on the above result, because (34) has two roots $\sigma^+_\omega < 1$ (corresponding to the above result) and $\sigma^-_\omega > 1$. Since with $\omega = \omega_2$ the matrix $L_\omega$ has only four real eigenvalues (see Figure 1) such that $\nu^+_1 > \nu^+_2 = \nu^-_2 = \omega_2 = 1 > \nu^-_1 > 0$, then for $\sigma_1 < 1$

$$\sigma^+_\omega = \frac{\nu_2}{\nu^+_1} = \frac{2}{1 + \sqrt{1 - \sigma_1}} - 1 < 1 \quad \text{and} \quad \sigma^-_\omega = \frac{\nu_2}{\nu^-_1} = \frac{2}{1 - \sqrt{1 - \sigma_1}} - 1 > 1.$$

But both $\nu_2$ and $\nu^-_1$ are subdominant eigenvalues and therefore the fact that $\sigma^-_\omega > 1$ has no practical significance.

The most interesting conclusion from this theorem is the fact that the minimum values of both spectral radius and subdominance ratio for the SOR iteration matrix are governed by the same formula (see (10) and (32)). In other words, for the same values of both $\rho(L_1)$ and $\sigma_1$ the quantities $\rho(L_\omega)$ and $\sigma_\omega$ achieve the same minimum value but with different values of $\omega$. It is evident that replacing $\rho(L_1)$, $\rho(L_\omega)$, and $E_t$ in Table 1 by $\sigma_1$, $\sigma^-_\omega$, and $E_t$ (defined by (28)), respectively, the data of this table illustrate also the efficiency of the power method in the asymptotic range as in the case of the SOR method.

Thus, the result of this theorem is of fundamental importance in the computational strategy for a "rapid" estimate of an "accurate" value of the optimum relaxation factor $\omega_{opt}$ in the SOR method.

The algorithm for determining $\omega_{opt}$, called the Sigma-SOR algorithm, is based on the following computational strategy. Assume that $A^*$ and $a^*$, appropriate estimates for $A_1 = \rho(L_1)$ and $\sigma_1$, respectively, are known. Using

$$\omega^* = \frac{2}{1 + \sqrt{1 - \sigma_1^* \lambda^*}},$$

we can obtain $\nu^* \equiv \rho(L_{\omega^*})$ by the power method iteration until a required convergence criterion is satisfied. Then from the relation (29) one obtains

$$\lambda_1 = \frac{1}{\nu^*} \left[ \frac{\nu^* + \omega^* - 1}{\omega^*} \right]^2,$$

which allows us to determine

$$\omega^*_1 = \frac{2}{1 + \sqrt{1 - \lambda^*_1}},$$

an a priori "accurate" estimate for $\omega_{opt}$. Thus, the accuracy of $\omega_{opt}$ is conditional to the computation of an accurate value of $\nu^*$.

As is demonstrated in numerical experiments given in the next section, the above algorithm, even with crude approximations $\lambda^*$ and $\sigma^*$, is very efficient and strongly competitive with the SOR adaptive procedure [1] when $\rho(L_1)$ is very close to unity $(0.999 < \rho(L_1) < 1)$.

Estimates for $\sigma^*$ approximating $\sigma_1$ can be obtained by observing the decay rate of some quantities, for instance

$$\sigma^{(t+1)} = \frac{\lambda^{(t+1)} - \lambda^{(t)}}{\lambda^{(t)} - \lambda^{(t-1)}},$$

or ratios of differences between the components of successive eigenvectors in the iteration process of the power method (19), (20), using a suitable norm (see, for example, [4], where the term dominance ratio is used for $\sigma$). As follows from
(18), for each \( j \)th nonzero component \( z_j \) of \( z \) approximating the eigenvector corresponding to the dominant eigenvalue in the power method, we have that

\[
\lambda^{(t+1)} = \frac{z_j^{(t+1)}}{z_j^{(t)}} = \lambda_1 \left[ \frac{a_1(u_1)_j + (e^{(t+1)})_j}{a_1(u_1)_j + (e^{(t)})_j} \right] \rightarrow \lambda_1 \quad \text{as} \quad t \to \infty,
\]

where

\[
(e^{(t)})_j = \sum_{i=2}^{n} \left( \frac{\lambda_i}{\lambda_1} \right)^t a_i(u_i)_j.
\]

Substituting (37) into (36), one obtains

\[
\sigma^{(t+1)} = \frac{a_1(u_1)_j + (e^{(t+1)})_j}{a_1(u_1)_j + (e^{(t)})_j} \left[ \frac{(e^{(t+1)})_j - 2(e^{(t)})_j + (e^{(t-1)})_j}{(e^{(t)})_j - 2(e^{(t-1)})_j + (e^{(t-2)})_j} \right],
\]

and for \( t \) sufficiently large, \( a_1(u_1)_j \gg (e^{(t)})_j \), so that

\[
\sigma^{(t+1)} \approx \frac{(e^{(t+1)})_j - 2(e^{(t)})_j + (e^{(t-1)})_j}{(e^{(t)})_j - 2(e^{(t-1)})_j + (e^{(t-2)})_j}.
\]

Assume now that for any \( t' \geq 1 \)

\[
\left( \frac{\lambda_2}{\lambda_1} \right)^{t'} a_2(u_2)_j \gg \left( \frac{\lambda_i}{\lambda_1} \right)^{t'} a_i(u_i)_j \quad \text{for all} \quad 3 \leq i \leq n.
\]

Equation (38) can be written in the form

\[
(e^{(t)})_j = \left( \frac{\lambda_2}{\lambda_1} \right)^t a_2(u_2)_j + (\tilde{e}^{(t)})_j,
\]

where

\[
(\tilde{e}^{(t)})_j = \sum_{i=3}^{n} \left( \frac{\lambda_i}{\lambda_1} \right)^t a_i(u_i)_j.
\]

Substituting (41) into (39) yields

\[
\sigma^{(t+1)} \approx \frac{\left( \frac{\lambda_2}{\lambda_1} \right)^t - 2 \left( \frac{\lambda_2}{\lambda_1} \right)^{t-1} + \left( \frac{\lambda_2}{\lambda_1} \right)^{t-2}}{\left( \frac{\lambda_2}{\lambda_1} \right)^t - 2 \left( \frac{\lambda_2}{\lambda_1} \right)^{t-1} + \left( \frac{\lambda_2}{\lambda_1} \right)^{t-2}} a_2(u_2)_j + (\tilde{e}^{(t+1)})_j - 2(\tilde{e}^{(t)})_j + (\tilde{e}^{(t-1)})_j
\]

But when \( t \gg t' \), the relation (40) implies that \( (\tilde{e}^{(t)})_j \) becomes sufficiently small, and it can be concluded that

\[
\sigma^{(t+1)} \to \frac{\lambda_2}{\lambda_1} = \sigma_1.
\]

In the calculation of \( \rho(\mathcal{L}_1) \) (or \( \rho(\mathcal{L}_\infty) \)) by means of the algorithm of the power method defined by (19)--(22), the notation \( \lambda_M \equiv \rho(\mathcal{L}_1) \) (or \( \nu_M \equiv \rho(\mathcal{L}_\infty) \)) corresponds to using the maximum norm \( \| \cdot \|_\infty \), and \( \lambda_E \equiv \rho(\mathcal{L}_1) \) (or \( \nu_E \equiv \rho(\mathcal{L}_\infty) \)) corresponds to using the Euclidean norm \( \| \cdot \|_2 \) in the scaling procedure.
With these notations,

\begin{align}
\sigma_M^{(t+1)} &= \frac{\lambda_M^{(t+1)} - \lambda_M^{(t)}}{\lambda_M^{(t)} - \lambda_M^{(t-1)}}, \\
\sigma_E^{(t+1)} &= \frac{\lambda_E^{(t+1)} - \lambda_E^{(t)}}{\lambda_E^{(t)} - \lambda_E^{(t-1)}}.
\end{align}

Usually, the convergence behavior of both $\lambda_M$ and $\sigma_M$ have a monotone decreasing character, whereas for $\lambda_E$ and $\sigma_E$ it was observed that they first increase and then (mainly for $\lambda_E$) slowly decrease as the number of iterations increases.

In the case of using the Euclidean norm for scaling purposes, the following two additional measures for $\sigma$ can be used:

\begin{equation}
\sigma_{EM}^{(t+1)} = \frac{\|y_E^{(t+1)}\|_\infty - \|y_E^{(t)}\|_\infty}{\|y_E^{(t)}\|_\infty - \|y_E^{(t-1)}\|_\infty},
\end{equation}

and

\begin{equation}
\sigma_{EE}^{(t+1)} = \frac{\|y_E^{(t+1)} - y_E^{(t)}\|_2 - \|y_E^{(t)} - y_E^{(t-1)}\|_2}{\|y_E^{(t)} - y_E^{(t-1)}\|_2 - \|y_E^{(t-1)} - y_E^{(t-2)}\|_2},
\end{equation}

where the successive eigenvectors $y_E^{(t+1)}$, $y_E^{(t)}$, $y_E^{(t-1)}$, and $y_E^{(t-2)}$ are generated by (19)–(22) with using the Euclidean norm for scaling.

As demonstrated in numerical experiments, the most rapid convergence is observed for $\sigma_{EE}$ with a monotone increasing character, which provides certain values estimating the true $\sigma_1$ from below.

As can be seen from Figure 1 the behavior of $\sigma_\omega$ near $\omega_2$ is similar in nature to the behavior of $\rho(\mathcal{L}_\omega)$ near $\omega_1$. From an inspection of the slope of the curve for $\sigma_\omega$ near $\omega_2$, it follows that errors with underestimating $\omega_2$ give larger values of $\sigma_\omega$ than errors (comparable in size) with overestimating $\omega_2$. In the range $1 < \omega < \omega_1$, the value of $\sigma_\omega$ can be determined from (33) in dependence on $\sigma_1$ and $(\omega - 1)/\nu_2$ (or $(\omega - 1)/\nu_1$ in the case of (33a)), and in the range $\omega_2 < \omega < \omega_1$, it is defined by $|\omega - 1|/\nu_1$.

Thus, from the viewpoint of obtaining the maximum rate of convergence in the power method, overestimating $\omega_2$ is less dangerous than underestimating $\omega_2$ by the same amount, but as $\sigma_1$ approaches unity, this becomes a more important problem because underestimating $\omega_2$ drastically decreases the rate of convergence.

On the other hand, however, underestimating $\omega_2$ may be attractive for accelerating convergence by the use of the Aitken $\delta^2$-process [5]. This procedure, known also under the name of Aitken extrapolation, is a useful tool for improving convergence, and can be used for any process converging linearly (i.e., as in (14), $z^t = \mathcal{F}z^{t-1}$). In the case of the simple power method, the convergent sequence $\{\lambda^{(t)}\}$ for the dominant eigenvalue can be transformed into a more rapidly convergent sequence $\{\hat{\lambda}^{(t)}\}$ by using

\begin{equation}
\hat{\lambda}^{(t)} = \lambda^{(t-2)} - \frac{(\lambda^{(t-2)} - \lambda^{(t-1)})^2}{\lambda^{(t-2)} - 2\lambda^{(t-1)} + \lambda^{(t)}}.
\end{equation}

This process will be most effective if both eigenvalues $\nu_1 \equiv \nu_1^+$ and $\nu_2 \equiv \nu_2^+$ are real and well separated from $\nu_3 \equiv \nu_3^+$. As can be easily concluded from Figure 1,
this occurs when \( \omega \) is close to \( \overline{\omega}_3 \), which minimizes \( \nu_3 \) for all \( 1 < \omega < \overline{\omega}_3 \) and provides the best separation of \( \nu_1 \) and \( \nu_2 \) from \( \nu_3 \). The distance of separation is a decreasing function as \( \omega \) increases for \( \overline{\omega}_3 < \omega < \overline{\omega}_2 \) and vanishes for \( \omega_2 < \omega < \omega_1 \) because in this region all subdominant eigenvalues have the same absolute value. Thus, the use of \( \sigma_{EE} \), providing an underestimated value of \( \sigma_1 \), can give some advantages in the form of an increased rate of convergence when the Aitken extrapolation is applied. This aspect will be discussed and illustrated by numerical results in the next section.

In conclusion it should be stated that in the efficient use of the power method for determining an accurate value of the optimum relaxation factor in the SOR iterative method, the relaxation factors \( \overline{\omega}_2 \) and \( \overline{\omega}_3 \) play an important role; \( \overline{\omega}_2 \) maximizes the rate of convergence in the simple power method, whereas \( \overline{\omega}_3 \), providing the best separation of two dominant eigenvalues from the remaining subdominant eigenvalues of the SOR iteration matrix, maximizes the rate of convergence of the Aitken extrapolation used as a practical technique for improving the convergence of the power method.

3. Numerical Experiments

In this section the results of numerical experiments are presented for the numerical solution of a two-dimensional elliptic equation of the form

\[
-D(x, y) \left[ \frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} \right] + \Sigma(x, y) \varphi = s(x, y) \text{ for } x, y \in \Omega
\]

with

\[
\varphi(x, y) = g(x, y) \quad \text{or} \quad \frac{\partial \varphi}{\partial n} = g(x, y) \text{ for } x, y \in \partial \Omega,
\]

where \( \Omega \) is an open bounded region with boundary \( \partial \Omega \), \( n \) is the exterior normal, \( D(x, y) > 0 \), and \( \Sigma(x, y) > 0 \).

The standard finite difference discretization of (46) in a spatial mesh imposed on \( \Omega \) leads to a system of linear equations of the form

\[
A\phi = b,
\]

where the components of \( \phi \) approximate the values of \( \varphi \) at each mesh point \( (x, y) \). In the case of the natural ordering of mesh points for the standard five-point difference operator, the \( n \times n \) coefficient matrix \( A \) has only five nonzero diagonals forming a tridiagonal block structure suitable for the implementation of the 1-line SOR algorithm, and is 2-cyclic consistently ordered [1].

Five test problems taken from the literature [6, 7] are considered with discontinuous coefficients \( D \) and \( \Sigma \), but chosen to be constant in each subregion \( \Omega_k \), and different boundary conditions on \( \partial \Omega \) for uniform and nonuniform mesh structures.

Test Problem 1. This example, obtained by assuming \( D = 1 \) and \( \Sigma = 0 \) in \( \Omega \), the unit square \( (0, 1) \times (0, 1) \), the Dirichlet boundary conditions \( \varphi = 0 \) on \( \partial \Omega \), is usually used as a model problem in the analysis of numerical solutions of elliptic-type problems. A square mesh with width \( h = \frac{1}{N+1} \) yields \( n = N^2 \) mesh points, which is also the order of \( A \). We assume \( n = 48 \times 48 = 2304 \), as in Problem A in [6].

Test Problem 2. In this problem (Problem B in [6]), whose domain and coefficients are depicted in Figure 2 (the numbers on the x-axis and y-axis in this
and subsequent figures are indices of mesh lines, not values of $x$ and $y$), there is a discontinuity of coefficients in the vertical direction, and mixed boundary conditions are used on $\partial \Omega$ as shown in Figure 2. The number of mesh points is $n = 96 \times 24 = 2304$, where $h = 1$ is assumed in both horizontal and vertical direction.

**Test Problem 3.** In this problem (Problem C in [6]), with $n = 24 \times 24 = 576$ and discontinuous coefficients, a nonuniform mesh is used. The mesh division, assumed the same in both horizontal and vertical direction, corresponds to the mesh division used in Problem 5 given in Reference 7 of [6]. The domain, coefficients and the mesh division are depicted in Figure 3.

**Test Problem 4.** This problem, taken from [7] (and analyzed in [8]), has a strongly discontinuous $D$, and $n = 48 \times 48 = 2304$ in the square mesh shown in Figure 4.
Test Problem 4. This problem, also taken from [7] (and analyzed in [8]), has a slightly modified mesh division, giving \( n = 42 \times 42 = 1764 \), in order to keep the number of horizontal lines divisible by 2 for convenient use of 2-line SOR algorithms. The domain, coefficients and mesh division (assumed the same in both directions) are depicted in Figure 5. In [7, 8] a uniform mesh with \( h = 0.05 \) was used, giving the number of mesh points \( n = 43 \times 43 = 1849 \).
For solving (47) in the above five test problems, the following line algorithms of the SOR iterative method are used [1, 6]:

1. SLOR—1-line system,
2. S2LOR—2-line system,
3. S2LCROR—2-line cyclically reduced system.

In our computations for each problem it was assumed that $s(x, h) = 0$ in (46), so that the unique solution of each discrete problem is the null vector. All components of the starting vector $\phi^{(0)}$ were equal to unity, and computations for each iterative method were continued until the maximum absolute value of all components of the iterate $\phi^{(i)}$ was less than a prescribed number $\varepsilon$. Thus, the stopping criterion

$$e^{(i)} = \|\phi^{(i)}\|_\infty \leq \varepsilon$$

can be considered as the most reliable measure of the error vector in estimating the accuracy of the solution.

All computations were carried out on a PC computer in single-precision FORTRAN for the SOR iteration (including the calculation of the coefficient matrices $A$), and in double-precision FORTRAN for the power iteration. The results of computation are shown in Table 2 (next page).

The accurate value of $\lambda_1$ was obtained with $\omega = 1$ when the stabilization to nine significant figures of $\lambda_E$ was observed in the power method ((19)-(22), using the Euclidean norm); $I_E$ and $I_A$ are the numbers of iterations observed in the power method without and with using the Aitken extrapolation (45), respectively; $I_S$ is the number of SOR iterations required to satisfy the stopping criterion (48) for two successive iterations with $\overline{\omega}_1$ as the optimum relaxation factor and for two values $\varepsilon = 10^{-6}$ and $\varepsilon = 10^{-8}$.

The results obtained when using the SOR adaptive subroutine [1, pp. 368–372] are shown under items 6, 7, and 8 of Table 2.

The data given in items 9–15 are related to computing the accurate value of $\nu_1$ (with stabilization to nine significant figures) in the power method with the value of $\omega = \overline{\omega}_2$ determined from (27a) where $\lambda_2 = \{\sigma_1[\text{accur}]\} \times \lambda_1$ and $\sigma_1[\text{accur}]$, approximated by $\sigma_{EE}$ (defined by (44d)), was obtained with the calculation of $\lambda_1$ in item 1 for $\omega = 1$. Hence, by (8) and (27), the accurate value of $\overline{\omega}_1$ can be found. Provided $\sigma_1$ is known, the accurate value of the optimum relaxation factor $\omega_{\text{opt}} \equiv \overline{\omega}_1$ can thus be efficiently computed. Comparison of the number of iterations $I_E$ (or $I_A$) given in items 2 and 13 allows us to illustrate the efficiency of the power method used in the case when $\omega = \overline{\omega}_2$ for each test problem. The values of $\sigma_{EE}$ given in items 14 and 15, and computed from $(\overline{\omega}_2 - 1)/\nu_1$ and (32), respectively, indicate the consistency of the results in all cases, except for Test Problem 5 solved by the SLOR iterative method, where $\sigma_1 = 0.9944$ was found to only four significant figures.

The results obtained for the Sigma-SOR algorithm are given in items 16–27. The subdominance ratio $\sigma_1$, approximated by $\sigma_{EE}$ is estimated once the stopping criterion

$$\delta^{(i)} = \left|\sigma_{EE}^{(i)} - \sigma_{EE}^{(i-1)}\right| \leq \delta = 10^{-3}$$

has been satisfied in two successive iterations in all test problems; $I_{EE}$ is the
<table>
<thead>
<tr>
<th>Problem</th>
<th>λ</th>
<th>L</th>
<th>ω</th>
<th>( I_e )</th>
<th>( I_e )</th>
<th>( I_e )</th>
<th>( I_e )</th>
<th>( I_e )</th>
<th>( I_e )</th>
<th>( I_e )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Computational results

<table>
<thead>
<tr>
<th>Test Problem 1</th>
<th>Test Problem 2</th>
<th>Test Problem 3</th>
<th>Test Problem 4</th>
<th>Test Problem 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLOR</td>
<td>SZLOR</td>
<td>SLCLOR</td>
<td>SLCLOR</td>
<td>SLOR</td>
</tr>
<tr>
<td>1. λ ( I_e ) (accur)</td>
<td>99815239</td>
<td>98372534</td>
<td>57834457</td>
<td>99815519</td>
</tr>
<tr>
<td>2. ( L_e )</td>
<td>984.85</td>
<td>551.67</td>
<td>370.42</td>
<td>236.91</td>
</tr>
<tr>
<td>3. ( \omega )</td>
<td>1.57327</td>
<td>1.77375</td>
<td>1.74344</td>
<td>1.79327</td>
</tr>
<tr>
<td>4. ( I_e )</td>
<td>0.86</td>
<td>72.64</td>
<td>0.269</td>
<td>189.160</td>
</tr>
<tr>
<td>5. ( I_e )</td>
<td>132</td>
<td>91.80</td>
<td>343</td>
<td>241.204</td>
</tr>
<tr>
<td>6. ( L_e )</td>
<td>1.83228</td>
<td>1.76273</td>
<td>1.76228</td>
<td>1.93597</td>
</tr>
<tr>
<td>7. ( I_e )</td>
<td>127.83</td>
<td>70</td>
<td>343</td>
<td>208.195</td>
</tr>
<tr>
<td>8. ( I_e )</td>
<td>155</td>
<td>98.80</td>
<td>431</td>
<td>275.234</td>
</tr>
<tr>
<td>9. ( L_e ) (accur)</td>
<td>.98770</td>
<td>.97572</td>
<td>.96760</td>
<td>.96167</td>
</tr>
<tr>
<td>10. ( L_e )</td>
<td>.99662</td>
<td>.98984</td>
<td>.96865</td>
<td>.99063</td>
</tr>
<tr>
<td>11. ( \omega )</td>
<td>1.7501</td>
<td>1.6861</td>
<td>1.6247</td>
<td>1.8235</td>
</tr>
<tr>
<td>12. ( \omega )</td>
<td>937281201</td>
<td>912084709</td>
<td>899887883</td>
<td>999810479</td>
</tr>
<tr>
<td>13. ( L_e )</td>
<td>120</td>
<td>101</td>
<td>82</td>
<td>80</td>
</tr>
<tr>
<td>14. ( I_e ) (comp)</td>
<td>.8003</td>
<td>.7303</td>
<td>.9499</td>
<td>.8327</td>
</tr>
<tr>
<td>15. ( I_e ) (theor)</td>
<td>.8003</td>
<td>.7304</td>
<td>.9499</td>
<td>.8327</td>
</tr>
</tbody>
</table>

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use
respective number of iterations required. As can be seen in Table 2, the above
stopping test provides an underestimation of \( \sigma_1 \) in all cases except for the
S2LCROR method in Test Problem 3, for which \( \sigma_{EE} \) gives a slight overesti-
mation of \( \sigma_1 \). In computing \( \bar{w}_2[\text{est}] \) according to (27a) it was assumed that
\( \lambda_2[\text{est}] = \sigma_{EE} \lambda_1^{(t)} \), where \( \lambda_1^{(t)} \) is the approximation of \( \lambda_1 \) obtained at iteration
t = \( I_{EE} \) and using Aitken extrapolation. In item 20 the value \( \nu_1 \) approximating
\( \nu_1 \) with \( \omega = \bar{w}_2[\text{est}] \) is obtained by satisfying the stopping criterion
\[
\delta^{(t)} = |\nu_1^{(t)} - \nu_1^{(t-1)}| \leq \delta = 10^{-8},
\]
which is achieved after \( I_E \) iterations without Aitken extrapolation. The corre-
sponding values of \( \lambda_E \) and \( \omega_E \) are given in items 22 and 23. In items 24–27
the same quantities are given when Aitken extrapolation is used. For the SLOR
method in Test Problem 1 there is a small difference between \( \omega_E \) and \( \omega_A = \bar{w}_1 \),
but in all remaining cases it is observed that \( \omega_E = \omega_A = \bar{w}_1 \) and \( I_A \) is smaller
than \( I_E \), as \( \bar{w}_2 \) is more underestimated by \( \bar{w}_2[\text{est}] \), because in this case the
separation of \( \lambda_1 \) and \( \lambda_2 \) from the remaining eigenvalues increases and Aitken
extrapolation becomes more efficient. In the case where the \( \omega \) used is close
to the true value of \( \bar{w}_2 \) (item 11), this separation of \( \lambda_1 \) and \( \lambda_2 \) from the re-
mainng eigenvalues disappears and the numbers of iterations \( I_E \) and \( I_A \) are
comparable (item 13).

Thus, with the choice \( \delta = 10^{-3} \) for \( \sigma_{EE} \) and \( \delta = 10^{-8} \) for \( \nu_1 \) and with
the use of Aitken extrapolation, the Sigma-SOR algorithm provides an estimate
for \( \omega_A = \bar{w}_1 \equiv \omega_{opt} \) to six significant figures in all considered test problems,
with \( I_{EE} + I_A \) (items 17 and 25) being the number of iterations required for
obtaining this estimate.

In all eigenvalue calculations carried out by means of the power method, all
components of the starting vector \( z^{(0)} \) were taken to be unity.

The behavior of \( \sigma_E, \sigma_M, \sigma_{EE} \), and \( \sigma_{EM} \) (defined by (44a, b, c, d)), represen-
ting different measures for \( \sigma_1 \), versus the number of iterations is depicted
in Figures 6–10 (see pp. 636–638) for all five test problems solved by means
of the SLOR iterative method. As can be seen in these figures, \( \sigma_{EE} \) converges
most rapidly to \( \sigma_1 \). (The true value of \( \sigma_1 \) given in item 9 of Table 2 is marked
in the figures by a straight line parallel to the \( x \)-axis.) In the initial phase of the
iteration process, \( \sigma_{EE} \) provides estimates of \( \sigma_1 \) from below, which are helpful
in using the Aitken extrapolation.

In the convergence behavior of \( \sigma_M \), the decreasing character is observed
as the number of iterations is increasing, but there are strong local variations
(occurring sometimes also for \( \sigma_{EM} \)) visible in all figures, except for Test Problem
2 depicted in Figure 7. In the case of Test Problems 1 and 4 (Figures 6 and 9),
it can be observed that for our starting vector \( z^{(0)} \), all of whose components are
equal to unity, all measures considered for \( \sigma_1 \) tend first to \( \lambda_3/\lambda_1 \) and then to
\( \sigma_1 = \lambda_2/\lambda_1 \) as the number of iterations increases. This is due to the fact that for
the assumed starting vector \( z^{(0)} \) the inequality \( a_3 \approx a_2 \) in the representation
(16) implies that in spite of \( \lambda_2 > \lambda_3 \), the inequality \( |a_5\lambda_3^2| \gg |a_2\lambda_1^2| \) holds
for appropriate “small” values of \( t \), so that the inequality (40) is not satisfied
because \( t < t' \) (where \( t \) may not necessarily be very small if \( t' \) is very large,
as occurs in the case of Test Problem 4) and \( \sigma^{(t)} \) will converge to \( \lambda_3/\lambda_1 \), the
dominant term in this range of \( t \)-values.
Figure 6. Test Problem 1

M: $\sigma_M$ (eq. (44a)); E: $\sigma_E$ (eq. (44b)); EM: $\sigma_{EM}$ (eq. (44c)); EE: $\sigma_{EE}$ (eq. (44d))

Figure 7. Test Problem 2

M: $\sigma_M$ (eq. (44a)); E: $\sigma_E$ (eq. (44b)); EM: $\sigma_{EM}$ (eq. (44c)); EE: $\sigma_{EE}$ (eq. (44d))
Figure 8. Test Problem 3

M: \( \sigma_M \) (eq. (44a)); E: \( \sigma_E \) (eq. (44b)); EM: \( \sigma_{EM} \) (eq. (44c)); EE: \( \sigma_{EE} \) (eq. (44d))

Figure 9. Test Problem 4

M: \( \sigma_M \) (eq. (44a)); E: \( \sigma_E \) (eq. (44b)); EM: \( \sigma_{EM} \) (eq. (44c)); EE: \( \sigma_{EE} \) (eq. (44d))
Moreover, it is interesting to notice that the convergence behavior of \( \sigma_{EE} \) and \( \sigma_M \) has a continuous character when passing from convergence to \( \lambda_3/\lambda_1 \) to convergence to \( \lambda_2/\lambda_1 \), whereas for \( \sigma_E \) and \( \sigma_{EM} \) strong deviations similar to discontinuities are observed.

It is a well-known fact that for the SOR iterative method the optimum relaxation factor \( \omega_{\text{opt}} \equiv \bar{\omega}_1 \) which maximizes theoretically the rate of convergence does not provide the best results. In practice, one observes the existence of a best relaxation factor \( \omega_B \) (slightly greater than \( \omega_{\text{opt}} \)) which minimizes the number of iterations for the required accuracy of the solution. Unfortunately, there is no rigorous analysis in the literature explaining the reasons for this \( \omega_B \) and predicting its value. From numerical experience, it can be concluded that \( \omega_B \) is a function of \( \omega_{\text{opt}} \) and the required degree of accuracy of the solution. One observes the following empirical formula:

\[
\ln(\omega_B - 1) = \frac{1}{c} \ln(\omega_{\text{opt}} - 1),
\]

where the correction coefficient \( c = 1.02 \) when using \( \varepsilon = 10^{-6} \), and \( c = 1.01 \) when using \( \varepsilon = 10^{-8} \), provides a quite satisfactory estimate for \( \omega_B \). The use of \( \omega_B \) obtained from the above formula allows us to improve the convergence. Usually, the number of iterations obtained with \( \omega_B \) is about 15% less than that obtained with \( \omega_{\text{opt}} \) for slowly convergent problems. The results obtained with \( \omega_B \) for two different stopping criteria are given in items 28–31.

The deterioration in the rate of convergence resulting from using an inaccurate value of \( \omega_{\text{opt}} \) is strongly dependent on the closeness of \( \rho(\mathcal{L}_1) \) to unity, and it seems to be reasonable that this dependence should be taken in consideration when estimating \( \omega_{\text{opt}} \) a priori. The nature of calculating \( \rho(\mathcal{L}_1) \) by

Figure 10. Test Problem 5

\( M: \sigma_M \) (eq. (44a)); \( E: \sigma_E \) (eq. (44b)); \( EM: \sigma_{EM} \) (eq. (44c)); \( EE: \sigma_{EE} \) (eq. (44d))
THE SIGMA-SOR ALGORITHM

means of the power method is such that the first few significant figures of \( \rho(\mathcal{L}_1) \) are rapidly fixed at the beginning of the power iterations, whereas convergence to the next figures begins to be governed by the subdominance ratio \( \sigma_i \). The behavior of \( \rho(\mathcal{L}_1) \) versus the number of power iterations for Test Problems 1 and 2 is depicted in Figure 11 where the dashed curves (denoted by 1a and 2a) correspond to using Aitken extrapolation for accelerating the convergence in the power method.

In the determination of \( \omega_{opt} \) based on a priori estimates for \( \rho(\mathcal{L}_1) \), the application of the stopping criterion

\[
\delta^{(t)} = |\lambda_1^{(t)} - \lambda_1^{(1-t)}| \leq \delta = 10^{-3}|(1 - \lambda_1^{(t)})|,
\]

where \( \lambda_1^{(t)} \) is an approximation of \( \lambda_1 \equiv \rho(\mathcal{L}_1) \) in the power iteration \( t \) using the Aitken extrapolation, yields results strongly competitive with the SOR adaptive procedure [1] when the values of \( \rho(\mathcal{L}_1) \) are close to unity.

In items 32-35 of Table 3 (next page) results are given for all test problems solved by the SLOR method in which the estimate of \( \omega_{opt} \) is based on the computation of \( \lambda_1 \equiv \rho(\mathcal{L}_1) \) by using the stopping criterion (52); the remaining items quoted from Table 2 are given for comparison purposes.

Table 4 summarizes the results obtained for different computational strategies implemented in four programs used for solving the test problems. The data given in this table represent the numbers of iterations required to obtain the solution which the stopping criterion \( \|\phi^{(t)}\|_\infty \leq 10^{-6} \) satisfied for two successive
Table 3. Results obtained with using the “dynamic” stopping criterion (52)

<table>
<thead>
<tr>
<th>Test Problem 1</th>
<th>Test Problem 2</th>
<th>Test Problem 3</th>
<th>Test Problem 4</th>
<th>Test Problem 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>λ₁ [accr]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ω₁</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I₁[c=10⁻⁶]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. λ₁ [accr]</td>
<td>.991815239</td>
<td>.998951986</td>
<td>.99961143</td>
<td>.9999893580</td>
</tr>
<tr>
<td>2. I₁</td>
<td>650</td>
<td>462</td>
<td>571</td>
<td>329</td>
</tr>
<tr>
<td>3. ω₁</td>
<td>1.83407</td>
<td>1.93728</td>
<td>1.98761</td>
<td>1.99193</td>
</tr>
<tr>
<td>4. I₁[c=10⁻⁶]</td>
<td>106</td>
<td>269</td>
<td>1347</td>
<td>2048</td>
</tr>
<tr>
<td>5. ω₁</td>
<td>1.83328</td>
<td>1.93587</td>
<td>1.98765</td>
<td>1.99186</td>
</tr>
<tr>
<td>6. ω₁</td>
<td>127</td>
<td>343</td>
<td>1853</td>
<td>3090</td>
</tr>
<tr>
<td>7. I₁[c=10⁻⁶]</td>
<td>39</td>
<td>46</td>
<td>22</td>
<td>25</td>
</tr>
<tr>
<td>8. I₁</td>
<td>100</td>
<td>67</td>
<td>76</td>
<td>69</td>
</tr>
<tr>
<td>9. ω₁</td>
<td>1.83704</td>
<td>1.93847</td>
<td>1.98785</td>
<td>1.99209</td>
</tr>
<tr>
<td>10. I₁[c=10⁻⁶]</td>
<td>99</td>
<td>229</td>
<td>1139</td>
<td>1736</td>
</tr>
<tr>
<td>11. I₁</td>
<td>35</td>
<td>96</td>
<td>155</td>
<td>101</td>
</tr>
<tr>
<td>12. I₁</td>
<td>1.83408</td>
<td>1.93662</td>
<td>1.98758</td>
<td>1.99191</td>
</tr>
<tr>
<td>13. I₁</td>
<td>106</td>
<td>283</td>
<td>1365</td>
<td>2077</td>
</tr>
<tr>
<td>14. I₁[c=10⁻⁶]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4. Comparison of computational strategies

<table>
<thead>
<tr>
<th>Method</th>
<th>Program No.</th>
<th>Test Problem 1</th>
<th>Test Problem 2</th>
<th>Test Problem 3</th>
<th>Test Problem 4</th>
<th>Test Problem 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-line</td>
<td>A1</td>
<td>127</td>
<td>343</td>
<td>1853</td>
<td>3090</td>
<td>1738</td>
</tr>
<tr>
<td></td>
<td>B1</td>
<td>106 (35)</td>
<td>283 (96)</td>
<td>1365 (155)</td>
<td>2077 (101)</td>
<td>1286 (18)</td>
</tr>
<tr>
<td></td>
<td>C1</td>
<td>99 (139)</td>
<td>229 (113)</td>
<td>1139 (98)</td>
<td>1736 (84)</td>
<td>1077 (49)</td>
</tr>
<tr>
<td>2-line</td>
<td>A2</td>
<td>83</td>
<td>208</td>
<td>1132</td>
<td>2047</td>
<td>1154</td>
</tr>
<tr>
<td></td>
<td>B2</td>
<td>72 (21)</td>
<td>193 (58)</td>
<td>866 (74)</td>
<td>1501 (55)</td>
<td>890 (8)</td>
</tr>
<tr>
<td></td>
<td>C2</td>
<td>66 (82)</td>
<td>160 (91)</td>
<td>740 (93)</td>
<td>1284 (69)</td>
<td>759 (80)</td>
</tr>
<tr>
<td></td>
<td>D1</td>
<td>61</td>
<td>169</td>
<td>752</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2-line</td>
<td>A3</td>
<td>70</td>
<td>195</td>
<td>997</td>
<td>1705</td>
<td>925</td>
</tr>
<tr>
<td>cyclically</td>
<td>B3</td>
<td>63 (17)</td>
<td>162 (45)</td>
<td>733 (70)</td>
<td>1270 (43)</td>
<td>775 (4)</td>
</tr>
<tr>
<td>reduced</td>
<td>C3</td>
<td>58 (82)</td>
<td>136 (87)</td>
<td>634 (75)</td>
<td>1103 (62)</td>
<td>654 (99)</td>
</tr>
<tr>
<td></td>
<td>D3</td>
<td>52</td>
<td>145</td>
<td>681</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

iterations. The numbers given in parentheses correspond to the number of iterations required to compute the relaxation factor ω for a given strategy.

The A program uses the SOR adaptive procedure [1]. In the B program the estimate of ωopt is based on computing λ₁ = ρ(Ł₁) by using the stopping criterion (52) and Aitken extrapolation as an acceleration procedure. The C program uses the Sigma-SOR algorithm for computing ωB. The numbers attached to the programs correspond to the applied solution methods, which are specified in the first column of the table. In addition, the results from [6] are quoted under the D2 program, which uses the 2-line cyclic Chebyshev method.
applied to the original system, and the D3 program, which uses the 2-line cyclic Chebyshev method applied to the cyclically reduced system. Both these programs were used in [6] for solving Test Problems 1, 2, and 3 only; the results from these programs for Test Problems 4 and 5 were not available.

4. CONCLUDING REMARKS

From the practical point of view, the best solution method is one that for the required accuracy provides the solution with the minimum total arithmetical effort, which is what mainly determines the cost of computations. In the case of the SOR iterative method, the arithmetical effort is roughly proportional to the number of SOR iterations required for obtaining the solution with a given degree of accuracy, and the number of power iterations required for estimating the appropriate relaxation factor $\omega$. Since the number of arithmetical operations per iteration in both SOR and power methods are comparable (the power method defined by (19)–(22) needs a few additional arithmetical operations for computing the Euclidean norm and for division by this norm), the efficiency of the assumed solution method can be measured in terms of the total number of iterations. Moreover, this total number of iterations, as well as the fraction of both SOR and power iterations, may change from problem to problem.

The number of SOR iterations is roughly inversely proportional to the rate of convergence where the deterioration of the convergence rate resulting from using an inaccurate value of $\omega_{\text{opt}}$ is strongly dependent on the closeness of $\rho(\mathcal{Z}_1)$ to unity. The speed of convergence in the power method is governed by the value of the subdominance ratio $\sigma_1$, which determines the rate of convergence, similarly as $\rho(\mathcal{Z}_1)$ does in the SOR method, and the number of power iterations is also strongly dependent on the closeness of $\sigma_1$ to unity or on the degree of separation of two dominant eigenvalues from the remaining ones, if the Aitken extrapolation is used. Thus, it seems that the selection and application of the iterative strategy for solving different problems should be based more on the analysis of results obtained in practice than on theoretical considerations.

In the test problems considered in this work and representing a class of nuclear engineering problems, we have

$$0.978 < \rho(\mathcal{Z}_1) < 0.99999 \quad \text{and} \quad 0.96 < \sigma_1 < 0.995,$$

so that the analysis of numerical results obtained for these problems should also be conclusive with solving large-scale scientific problems.

It seems that in the selection of computational strategy in solving elliptic-type problems, the SOR adaptive technique (implemented in the A1, A2, and A3 programs) is favored in the literature [1, 2, 3, 4, and 6] as a more efficient solution method in comparison with the computational strategy based on a priori estimate of $\omega_{\text{opt}}$. However, the numerical experiments on all test problems considered here show that the B2, B2, and B3 programs, in which an a priori estimate for $\omega_{\text{opt}}$ is obtained by calculating $\lambda_1 \equiv \rho(\mathcal{Z}_1)$ with the power method accelerated by Aitken extrapolation and using the stopping criterion (52), are competitive with the A1, A2, and A3 programs, especially when $\rho(\mathcal{Z}_1)$ is close to unity.

As can be seen in Table 4, in the case of Test Problem 1 the B1 program
needs 14 iterations more (that is, about 10% more) than the A1 program. But for Test Problem 4 the difference is equal to 912 iterations in favor of the B1 program, which corresponds to about 40% more iterations in the A1 program. Since both test problems have the same size (2304 mesh points), the advantages resulting from solving Test Problem 4 by the B1 program in comparison to the A1 program can be estimated by this difference of iterations, which in this case is about seven times greater than the total number of iterations required for solving Test Problem 1 by the A1 or B1 programs.

Suppose that both problems are solved with an a priori estimate for \( \omega_{opt} \) based on using the accurate value of \( \rho(\mathcal{L}) \) given in item 1 of Table 3 and obtained with 650 and 329 iterations (item 2 of Table 3) for Test Problems 1 and 4, respectively. Then, in the case of Test Problem 1 the solution is obtained with 106 iterations (the same number of iterations as for the B1 solution), but the total number of iterations is increased to 755, that is, 615 iterations more than for the B1 solution given in Table 4. For Test Problem 4 the total number of iterations (accompanied by a small decrease of SOR iterations) is increased to 2377, that is, 199 iterations more than for the B1 solution but still much less than for the A1 solution. A similar behavior can be observed when comparing the results of Table 4 given for the A2 and A3 programs with those given for the B2 and B3 programs, respectively.

From the above comparisons, it is apparent that in the solution method based on a priori estimates for \( \omega_{opt} \), the main difficulty lies in the choice of the degree of accuracy appropriate for estimating \( \rho(\mathcal{L}) \) in a given problem; it is probably for this reason that a priori estimates for \( \omega_{opt} \) are given less attention in the literature. However, as can be concluded from the results given in Table 4 for the B1, B2, and B3 programs, the simple trick of using the stopping criterion (52) conditioned by the closeness of \( \rho(\mathcal{L}) \) to unity allows us in some sense to avoid this main difficulty and to make a priori estimation of \( \omega_{opt} \) a more useful computational technique and competitive with the solution method based on using the SOR adaptive procedure [1], especially for problems in which the values of \( \rho(\mathcal{L}) \) are very close to unity. In the range \( 0.98 \leq \rho(\mathcal{L}) \leq 0.999 \), represented by Test Problems 1 and 2, the SOR adaptive procedure discussed extensively and illustrated numerically in [1] just for this range of values of \( \rho(\mathcal{L}) \), provides solutions with a smaller number of iterations than in the case of using a priori estimates for \( \omega_{opt} \) based on the stopping test (52). But as was demonstrated above for Test Problem 1, the advantages resulting from decreasing the total number of iterations have no practical significance because in this range of spectral radii, the deterioration of the convergence rate caused by using an inaccurate value of \( \omega_{opt} \) does not strongly change the number of iterations. For the class of problems with \( 0.999 < \rho(\mathcal{L}) < 0.99999 \), represented by Test Problems 3, 4, and 5, the efficiency of solution becomes more sensitive to the accurate value of \( \omega_{opt} \) as \( \rho(\mathcal{L}) \) approaches unity, and the computational strategy based on determining an accurate value of \( \omega_{opt} \) prior to the SOR solution is much superior than the SOR adaptive technique, as can be seen in Table 4. In this case, the last estimate for \( \omega_{opt} \) in the SOR adaptive technique is most time-consuming because \( \sigma_\omega \) becomes close to unity (see Figure 1). It is interesting to note that in the case of Test Problem 5 extremely small numbers of iterations are required to a priori estimate \( \omega_{opt} \) in the B1, B2, and B3 programs.
In the C1, C2, and C3 programs, the Sigma-SOR algorithm defined by (35a)-(35c) is used for the a priori determination of $\omega_{opt}$, whose value to six significant figures was computed with the choice of $\delta = 10^{-3}$ for approximating $\sigma_1$ by $\sigma_{EE}$ and $\delta = 10^{-8}$ for approximating $\nu^*$ by $\nu_A$, and using the Aitken extrapolation. The detailed results are given in items 16–27 in Table 2. In SOR iterations the best relaxation factor $\omega_B$ is used which is computed from the relation (51) and is given in item 28 of Table 2. As can be seen in Table 4, the Sigma-SOR algorithm needs about 100 iterations for computing $\omega_{opt}$ to six significant figures in all test problems. For Test Problem 1 the number of iterations required to obtain this accurate estimate for $\omega_{opt}$ exceeds the number of SOR iterations, so that the total number of iterations in the C1, C2, and C3 programs is about two times greater than in the A1, A2, and A3 programs, respectively. However, as $\rho(\mathcal{L}_1)$ becomes close to unity in the next test problems, the efficiency of the computational strategy with the Sigma-SOR algorithm is strongly improving in comparison to the former solution methods. Moreover, it is observed that in the case of Test Problems 3, 4, and 5 solved by the C1, C2, and C3 programs, the total number of iterations (needed for estimating $\omega_B$ and obtaining the solution) is smaller than the number of SOR iterations observed when using the accurate value of $\omega_{opt} = \omega_1$ (items 3 and 4 in Table 2).

The results for Test Problems 1, 2, and 3 obtained in [6] by means of the D2 program, using the 2-line cyclic Chebyshev method applied to the original system, and the D3 program, using the 2-line cyclic Chebyshev method applied to the cyclically reduced system, are given additionally in Table 4. From an inspection of these results, it is apparent that the solution efficiency of the D2 and D3 programs, which is the best in the case of Test Problem 1, decreases when going to Test Problems 2 and 3 in comparison to the convergence behavior of the C2 and C3 programs, respectively. For Test Problem 3, the C2 and C3 programs provide solutions with the total number of iterations somewhat greater than in the D2 and D3 programs. However, as follows from an exact calculation of the number of arithmetical operations for the obtained solutions, the C2 and C3 programs need somewhat less total arithmetical effort than the D2 and D3 programs, respectively. This is due to the fact that in each iteration of the D2 and D3 programs, except for the arithmetical operations related with the solution, additional arithmetical operations are required for the computation of the Euclidean norm, whereas in the C2 and C3 programs only about 10% of the number of iterations (the numbers given in parentheses in Table 4) are related to those additional computations.

Thus, it can be concluded from the results obtained for our test problems, that the Sigma-SOR algorithm based on the important theoretical result given by (32) is a useful computational tool for the calculation of an accurate a priori estimate of $\omega_{opt}$, which in turn allows to determine the best relaxation factor $\omega_B$ from (51) when solving problems for which $0.999 < \rho(\mathcal{L}_1) < 1$. In comparison to the SOR adaptive procedure, the efficiency of the Sigma-SOR algorithm increases as $\rho(\mathcal{L}_1)$ and $\sigma_1$ become closer to unity; and it seems that for the range $0.999 < \sigma_1 < 1$, the Sigma-SOR algorithm should be extremely efficient. In the case when the matrix problem (47) is to be solved many times for different vectors $b$, the advantages resulting from using $\omega_B$ obtained by means of the Sigma-SOR algorithm are obvious.
Finally, it should be mentioned that the subsequent updated values of \( \omega_i \) in the SOR adaptive technique are underestimated with respect to \( \omega_{\text{opt}} \), but this underestimation drastically decreases the rate of convergence as \( \rho(\mathcal{L}_1) \) becomes close to unity, and therefore the efficiency of the SOR adaptive procedure also decreases when \( \rho(\mathcal{L}_1) \) approaches unity.

ACKNOWLEDGMENT

The author would like to thank Drs. J. Kubowski and K. Pytel for their useful discussions and comments, as well as M. Sci. P. Jarzembski for his expert programming assistance. Thanks are also due to the Editor, Professor W. Gautschi, for his significant contribution with revising the manuscript.

BIBLIOGRAPHY


Institute of Atomic Energy, 05-400 Otwock-Swierk, Poland
E-mail address: r05zw@cyf.gov.pl