JOURNAL OF COMPUTATIONAL PHYSICS 32, 403–419 (1979)

A Simultaneous Coordinate Relaxation Algorithm for Large, Sparce Matrix Eigenvalue Problems*

RICHARD C. RAFFENETTI

Theoretical Chemistry Group, Chemistry Division, Argonne National Laboratory,[†] *Argonne, Illinois 60439, and Institute for Computer Applications in Science and Engineering, M/S 132-C, NASA Langley Research Center, Hampton, Virginia 23665*

Received April 21, 1978; revised September 6, 1978

A new algorithm for simultaneous coordinate relaxation is described. For the determination of several extreme eigenvalues and eigenvectors of large, sparse matrices the simultaneous algorithm affords significant advantages in comparison with a coordinate relaxation algorithm applied to determine individual eigenvalues and eigenvectors in turn. Results of application of the algorithm to test matrices are discussed.

INTRODUCTION

Relaxation methods have been shown to be very effective for solving certain matrix eigenvalue problems [1–3]. Characteristic of these problems are eigenvectors which have a few dominant components and many secondary ones. This type of structure is encountered when a matrix eigenproblem is generated to solve a physical problem by a "basis set" approach and when moreover a few terms of the basis form a good approximation to the eigenfunction sought. Although direct methods are most effective for the computer solution of eigenproblems where the matrices can be contained in the fast memory, other methods must be used when the matrices are too large. Direct methods modify the matrix elements, whereas when a matrix is sparse, which is a frequent characteristic of a basis set approach, that feature is undesirable.

In comparison with sequential solution of individual eigenvectors and eigenvalues by coordinate relaxation [3], the application of the iterative *simultaneous* relaxation method described here to determine the several most dominant eigenvectors and eigenvalues of a matrix can be beneficial in two ways. Because the single vector scheme displays slowed convergence when there are degenerate (or near degenerate) eigenvalues, it is reasonable to expect that a method which converges instead on the subspace containing the multiple roots will not be slowed. Such a procedure is the basis of the scheme called "simultaneous iteration", a multiple-vector generali-

* Work performed under the auspices of the Division of Physical Research of the U.S. Department of Energy. The U. S. Government's right to retain a nonexclusive, royalty-free license in and to the copyright covering this paper, for governmental purposes, is acknowledged.

[†] Present address.

0021-9991/79/090403–17$02.00/0

zation of the power method which was proposed by F. L. Bauer [4]. Second, if the matrix is so large that it must be held "out of core", one can benefit by a reduction in the quantity of "*I/O*", and consequently a reduction of "wait time" which can represent a sizeable portion of the computation charge associated with the problem solution.

Simultaneous vector methods have an advantage when the matrix may have special characteristics. In quantum chemistry configuration interaction calculations, in which such large, sparse matrices are formed and solved for the few lowest eigenvalues and their eigenvectors, the physical situation may cause a matrix to be effectively or actually blocked (i.e. no non-zero, off-diagonal elements exist which would connect large diagonal blocks of the matrix). This occurs for a molecular system which approaches a dissociation limit. At some point the fragments, treated as a single system, become independent, first in an effective sense when the off-diagonal block elements exist but are very small, and then in actuality, as the elements disappear. In this case initial vectors are quite important. The matrix blocking will not allow correction of vectors to incorporate components from all blocks unless the original vector has such components. As a result, the eigenvectors and eigenvalues obtained may be from the wrong matrix block and hence would not correspond to the desired solutions. Simultaneous solution for several vectors is likely to introduce the appropriate space in which the correct vectors can be found.

On the other hand, there are potential disadvantages. The usual algorithms require a solution vector in core in addition to an associated work space of about equal size. The simultaneous approach therefore puts quite a demand on the memory resource because more solution vectors and more work spaces are active. Also, more operations are required per iteration, so that if convergence is not improved, then there may be more total operations. The effects of these disadvantages should be kept in mind when constructing a computer program to implement the present algorithm.

In the following, the relaxation method is described briefly in order to provide the background for the present scheme. A variation of it due to Shavitt *et al.* [3] and known as "root-shifting", which is used to obtain subdominant eigenvalues and eigenvectors in sequence, is then introduced. The simultaneous relaxation scheme, which is a multiple-vector generalization of the root-shifting scheme, is then described in detail. Thereafter follow the results of tests run with the basic algorithm.

The Method of Coordinate Relaxation for an Extreme Eigenvalue

The problem is to find λ and x, an extreme eigenvalue and eigenvector, given the matrix A, where

$$Ax = \lambda x. \tag{1}$$

In the relaxation method, the best current estimate of x is replaced by itself plus some multiple of another unit vector u, i.e.

$$x \leftarrow x + \alpha u. \tag{2}$$

The relaxation parameter α is described below.

Solution of (1) for a dominant eigenvalue is equivalent to extremizing the Rayleigh quotient

$$\rho(x) = (x^TAx)/(x^Tx). \tag{3}$$

If u is a coordinate vector, $u = e_i$, $(e_i)_j = \delta_{ij}$ $(i, j = 1, 2,..., n)$, one easily obtains [3] the quadratic equation

$$a\alpha^2 + b\alpha + c = 0 \tag{4}$$

with

$$\begin{aligned} a &= (A_{ii}x_i - \phi_i)/(x^Tx) \\ b &= A_{ii} - \rho(x) \\ c &= \phi_i - \rho(x)\, x_i \end{aligned} \tag{5}$$

and where

$$\phi_i = a_i^Tx \equiv e_i^TAx. \tag{6}$$

Stable solution of the quadratic in (4) to determine α for each coordinate direction in turn $(i = 1, 2,..., n)$ is the process called "optimal" coordinate relaxation (CR). As the eigenvector is neared, a simple linear approximation to the solution of (4), which is equivalent to the Nesbet algorithm [1], provides adequate improvement [2]. However, if the initial guess for x is not reasonable, too early introduction of such an approximation may be deleterious to the convergence. The variation of CR in which α is taken to be a fixed multiple of the solution of (4) is called "over-relaxation" [5, 6]. Choice of an optimal overrelaxation factor is likely to be strongly problem dependent.

An *adequate* initial guess for x is the coordinate vector corresponding to the extreme diagonal element of A. To choose the sign of the radical in the solution of (4), one must only be sure that the value of $\rho(x)$ tends to the appropriate extreme (minimum or maximum) and convergence is thereby guaranteed. A better initial guess for x is obtained by extracting a submatrix of A corresponding to *several* of the most extreme diagonal elements and computing its lowest eigenvector. If x is not close to a coordinate vector, this choice will be substantially better.

The basic CR algorithm is summarized as follows:

```
initialize: determine $x$; compute $x^TAx$, $x^Tx$, $\rho(x)$
test for convergence: if satisfied, then stop, else
   begin the row iteration: $i = 1, 2,..., n$
       obtain the $i$th row of $A$
       determine $\alpha$.
       $x \leftarrow x + \alpha e_i$
       update $x^TAx$, $x^Tx$, $\rho(x)$
   end the row iteration
return to the convergence test.
```

An important feature of relaxation schemes, as first described by Nesbet [1], is that certain quantities can be updated rather than recomputed. Note also that it is not necessary to iterate the row index in any particular order. The optimal order is probably that one dictated by the sequence of successively decreasing values of α, but for matrices held out of core utilizing that order generally does not lead to a practical algorithm. To compute each α is an expensive part of the process, so that searching for the largest α would be highly inefficient, considering the possible gain. The usual procedure is to process the rows in some predetermined order, that being determined largely by the order in which A is stored and accessible.

Coordinate Relaxation with Root-Shifting for Near-Extreme Eigenvalues and their Vectors

To obtain the $(k + 1) - st$ lowest (or highest) eigenvalue and its eigenvector by this algorithm one must have obtained the k eigenvectors corresponding to the k eigenvalues which are lower (or higher, respectively). The matrix is then modified so that all of the latter eigenvalues are shifted to a value which makes them less extreme, so that the desired one, the $(k + 1) - st$, is left as the lowest (or highest) eigenvalue of the modified matrix [3]. The equation is

$$A^{(k+1)} = A - X_k M X_k^T \tag{7}$$

where M is the diagonal matrix of order k,

$$M_{ij} = \delta_{ij}(\lambda_i - \sigma_i), \qquad (i, j = 1, 2,..., k) \tag{8}$$

and X_k is the $n \times k$ matrix containing the k eigenvectors of length n which correspond to the eigenvalues $\lambda_1, \lambda_2,..., \lambda_k$. The scalars σ_i are the desired new eigenvalues of the modified matrix $A^{(k+1)}$ which correspond to the same eigenvectors, and are chosen so that they are definitely higher (or lower) than the one which is to be found. Note that a useful bound for the position of λ_{k+1} in the spectrum of A is available as $\rho(x_{k+1})$, determined from the initial vector x_{k+1}, orthogonalized to $x_1 \cdots x_k$. Note also that $A^{(k+1)}$ is $n \times n$, it has the same eigenvectors as A, and the eigenvalue corresponding to x_i is either σ_i (for $i \leqslant k$) or λ_i (for $i > k$). (A common variant of this spectral shift is known as deflation: $\sigma_i = 0$, $i = 1, 2,..., k$. The suitability of this latter choice in connection with CR rests upon the position of zero in the spectrum of A.)

Effective and efficient implementations of CR and CR with "root-shifting" (CR/RS) have been described by Shavitt *et al.* [3]. The key to the computational success of the CR/RS algorithm is the fact that the matrix $A^{(k+1)}$ is not constructed explicitly. That process would destroy the advantage associated with the sparseness of A, in addition to its being a computationally tedious process. Equation (7) may be used to derive the equations which are analogous to Eqs. (3–6). The quantities $A_{ii}^{(k+1)}$, $\phi_i^{(k+1)}$ and

$\rho^{(k+1)}(x)$ are related to those in Eqs. (5–6) by the addition of a second term which derives from the second term in Eq. (7). This new term is simple as long as k is small, say of the order of ten or less. The quantities x^Tx, x^TAx, and $\rho(x)$ are most effectively obtained by updating. The necessary equations are given by Shavitt *et al.* [3].

Simultaneous Coordinate Relaxation

The generalization of coordinate relaxation for the simultaneous determination of a few (p) eigenvectors, $X_p = (x_1, x_2, ..., x_p)$, and eigenvalues is based on a CR/RS procedure. In order to obtain the correction for the $(k + 1)$ *st* vector $(k + 1 \leqslant p)$ one can proceed as if X^k *were* known and use the same equations as are used to obtain X_{k+1}. However, in the prescribed algorithm, the group of vectors Y_p which *are* known at any given moment are non-orthogonal, i.e.

$$Y_p{}^T Y_p \equiv Q \neq I \tag{9}$$

The space subtended by Y_p may be expressed in terms of an orthogonal set of vectors X_p by the relation

$$X_p = Y_p C \tag{10}$$

and it follows that

$$CC^T = Q^{-1}. \tag{11}$$

The vectors are not orthogonal, and furthermore X_p is not even a collection of eigenvectors in general. Therefore it is necessary to consider what the effect will be on the matrix shift given by Eq. (7) and (8) which require knowledge of the eigenvectors.

The vectors Y_k are approximations to the first k eigenvectors with Rayleigh quotients $\rho(y_i)$. They span a k-dimensional subspace (k-space) of the entire n-dimensional space. The generalization of Eq. (7) necessarily consists of modifying A such that the eigenvalues of the entire k-space are shifted so that the Rayleight quotient $\rho(y_{k+1})$ is left in the extreme position. To effect this, M in (7) is limited to the form

$$M = -\sigma_{k+1} I \tag{12}$$

where the scalar σ_{k+1} is chosen appropriately (*vide infra*). This important choice permits the following simplification of Eq. (7):

$$A^{(k+1)} = A + \sigma_{k+1} X_k X_k{}^T \tag{13}$$

(Note that this equation is invariant to an orthogonal transformation among the orthogonal vectors which comprise X_k). Unfortunately, at any given moment during the eigenvector determination, only Y_k is available, so that the working equation becomes

$$A^{(k+1)} = A + \sigma_{k+1} Y_k Q^{-1} Y_k^T \tag{14}$$

upon substitution of (10) and (11). It would not be practical to compute Q^{-1} to determine the relaxation parameters; it does not appear possible to update Q^{-1} simply either. Therefore, it would seem that to approximate Q^{-1} is the only viable alternative. The two approximations which have been used in this work are:

$$Q^{-1} \simeq I \tag{15}$$

and

$$Q^{-1} \simeq 2I - Q \tag{16}$$

The first approximation derives from the fact that at the start of each iteration one normally begins with orthogonal vectors so that $Q = I$. The second approximation comprises the first terms in the binomial expansion of $(I + \Delta)^{-1}$ where $Q = I + \Delta$. In practice, as demonstrated by the results involving certain test matrices, there appeared to be a negligible difference in the eigenvector convergence rates between these alternatives. The second approximation is expected to be a bit more generally useful, i.e., applicable to a larger class of matrix eigenproblems, and correspondingly represents an implementation which is a bit more expensive. Whether or not the more complicated approximation is necessary is probably problem dependent.

Equation (4) is solved to provide the vector update in each coordinate direction, with Eqs. (5) and (6) defining the coefficients. The quantities

$$\begin{aligned} A_{ii}^{(k+1)} &= A_{ii} + \sigma_{k+1} \sum_{rs}^{k} Y_{ir} Q_{rs}^{-1} Y_{is}, \\ \phi_i^{(k+1)} &= \phi_i + \sigma_{k+1} \sum_{rs}^{k} Y_{ir} Q_{rs}^{-1} Q_{s,k+1}, \end{aligned} \tag{17}$$

and

$$\rho^{(k+1)}(y_{k+1}) = p^{(k+1)}(y_{k+1})/(y_{k+1}^T y_{k+1})$$

with

$$p^{(k+1)}(y_{k+1}) = p(y_{k+1}) + \sigma_{k+1} \sum_{rs}^{k} Q_{r,k+1} Q_{rs}^{-1} Q_{s,k+1}$$

are substituted for A_{ii}, ϕ_i, and $\rho(x)$. Equations (17) are a consequence of substitution of (14) for A in the definitions of the latter quantities. The matrices P and Q are easily updated using

$$P_{kl} \leftarrow P_{kl} + \alpha_k \phi_i(y_l) + \alpha_l \phi_i(y_k) + \alpha_k \alpha_l A_{ii}$$

and

$$Q_{kl} \leftarrow Q_{kl} + a_k y_{il} + \alpha_l y_{ik} + \alpha_k \alpha_l$$

where i is the current coordinate direction.

The shift parameter σ_{k+1}, as in the CR/RS procedure, must be chosen so as to shift the first k eigenvalues of A beyond the $(k+1) - st$ by an amount sufficient so that in $A^{(k+1)}$, the shifted eigenvalues do not lie between the new first and second most extreme eigenvalues. Too large a shift can degrade the precision of the elements of $A^{(k+1)}$, and so it is desirable to shift by an amount just sufficient to satisfy the former requirement. An effective program will update the shift parameters during the course of the iterations as the energy spacings become known. Knowledge of the energy spacings may also be used to provide the initial shift parameters. Otherwise a convenient initial choice for σ_{k+1} which has been employed in the test runs is

$$\sigma_{k+1} = \frac{2}{3} [\rho(y_{k+1}) - \rho(y_1)] \cdot \left(\frac{2k+3}{k}\right). \tag{18}$$

The form of this choice derives from an assumption of approximately equal spacing of eigenvalues between λ_{k+1} and λ_1 plus a factor of 2/3 to take care of significant deviations.

As mentioned above, during the eigenvalue determination the vectors Y_p are not orthogonal. Although it is possible to provide for this in the equations for the determination of the relaxation parameters (*vide supra*), it is desirable to have the vectors orthogonal and also to have the best Rayleigh quotients corresponding to the currently known A subspace, $Y_p{}^T A Y_p$. These desires are easily satisfied by solving the p-dimensional generalized eigenvalue problem

$$PC = QC\Lambda \tag{20}$$

with

$$P = Y_p{}^T A Y_p \tag{21}$$

and Q as given in Eq. (9). Then, by Eq. (10), the set of orthogonal vectors X_p is easily produced from Y_p. Moreover, the diagonal matrix Λ contains the p Rayleigh quotients which correspond to the vectors in X_p.

Equation (20) may be solved at any time which is convenient, or when it is necessary so that solution of the other relaxation equations will be *effective*. (This could be determined by monitoring the growth of the elements of the matrix Δ.) In practice it is convenient to carry out this substep at the end of each major iteration. The amount of machine time necessary for correcting the vectors in this way is of the order np^2, which represents the matrix multiplication of Eq. (10). Solution of Eq. (20) is negligible for the case where $n \gg p$. Since a step of the order np^2 cannot be considered negligible with respect to solving the rest of the relaxation equations, the question of just how often one should carry out this process is of considerable interest. Test results are given below which show how the frequency of the generalized eigenproblem solution affects the convergence of this iterative scheme.

The SCR Algorithm

The complete iterative algorithm consists of "major" and "minor" cycles. In a major cycle all of the coordinate directions, $i = 1, 2, \ldots, n$, are treated in some order. A minor cycle consists of changing each of the vectors $y_1, y_2, \ldots, y_p$, in that order, for the ith coordinate. The change in y_{k+1} is determined by use of (4), where $Y_k = (y_1, y_2, \ldots, y_k)$, *viz.* only the current vectors corresponding to the more extreme eigenvalues are used to form $A^{(k+1)}$. It is clear that the equations used to obtain changes for $y_1 (k = 0)$ are just the ordinary CR equations which employ $A^{(1)} \equiv A$ and do not require vectors other than $y_1 \equiv x_1$. Because of this, the most extreme eigenvector will converge at least as quickly as it would in an equivalent nonsimultaneous CR scheme. With regard to comparison with a nonsimultaneous version of CR, the equations solved to obtain $\alpha_1, \alpha_2, \ldots, \alpha_p$ in the ith minor iteration all involve only the ith row of A by virtue of (6). Therefore, when a row of A is brought into the work area, a quantity of computing of the order of p times as many operations is carried out before another row is needed. This is a significant advantage of the present method; it is seen that the overall input of A is decreased by a factor equal to the number of vectors which are iterated simultaneously. This is, of course, assuming that the number of row iterations times the number of vectors iterated simultaneously is not greater than the total number of row iterations required in the CR/RS procedure; in fact it is likely to be much less as the results below demonstrate.

The algorithm is summarized in a free form as follows:

```
initialize: determine $X_p$; compute $P$, $Q$, $\rho_j$ $(j = 1, 2, \ldots, p)$
test for convergence: if satisfied, then stop, else
  begin the row iteration: $i = 1, 2, \ldots, n$
    obtain the $i$th row of $A$
    compute $a_i^T x_j$ $j = 1, 2, \ldots, p$
    begin the column iteration: $j = 1, 2, \ldots, p$
      $x_j \leftarrow x_j + \alpha e_i$
      update $P$, $Q$, $\rho_j$
    end the column iteration
  end the row iteration
  solve: $PC = QC\Lambda$
  $X_p \leftarrow X_p C$
  update $P \leftarrow \Lambda$, $Q \leftarrow I$, $\rho_j = \Lambda_{jj}$ $(j = 1, 2, \ldots, p)$
return to the convergence test
```

Note that the inner products of the rows of A and the p vectors x_j are computed outside of the column iteration loop. Ostensibly there is no difference in the amount of computing necessary if the inner products were computed inside the loop, but a careful simultaneous organization can minimize the movement of data to and from the registers.

In order to effectively compare the present method with others it is useful to summarize the operation counts for the dominant computations. Following the usual practice, the quantities given represent the numbers of multiplications per major iteration. The dominant term (normally) represents the inner products and is given by $n^2p\ d(A)$ where $d(A)$ is the density of non-zero elements in A. Updating of P and Q within the row iteration is represented by about $2np^2$ and the determination of the α's is βnp where β is a constant representing the operations to find the square root. Following the row iterations, terms are necessary to represent the generalized eigenproblem and the associated vector changes and updates. Replacing X_p by X_pC requires np^2 multiplications, while solution of the eigenproblem requires no more than about $2p^3$. The total operational cost of an iteration of SCR is:

$$n^2p\ d(A) + 2np^2 + \beta np + fp^2(n + 2p) \tag{22}$$

The group of terms multiplied by f, a frequency factor, are those associated with the generalized eigenproblem. Since the np^2 term is possibly non-negligible if $d(A)$ is very small, it may be worthwhile to employ a frequency smaller than unity if there is not a large effect on the convergence rate. The effect has been observed empirically for the limited tests described below. Conclusions regarding the frequency are given.

As described prior to the algorithm, compared to the CR/RS procedure, the total amount of input operations, consisting essentially of those necessary to have access to the rows of A, may be reduced substantially. The amount of computer memory required for the basic algorithm is largely np, the space to store X_p. If matrix symmetry is used, according to the method of Shavitt [2], an additional memory area of np words is required to maintain the column sums $U_p = (u_1, u_2, ..., u_p)$; when X_p is replaced by X_pC, the column sums are replaced in the same way: $U_p \leftarrow U_pC$. This requires an additional fnp^2 multiplications.

Results

The tests described here show very well the strength of this algorithm in comparison with the non-simultaneous version of root-shifted coordinate relaxation. All results described were carred out on two matrices of dimension $N = 50$. As regards the use of a small dimension for testing, it is found that convergence rate cannot be tied directly to matrix dimension but it is more a question of matrix conditioning. Enlargement of the matrices described below in a straightforward way produced new matrices for which the convergence was not significantly slower. Enlargement does change the conditioning of the matrix.

The first test matrix employed is referred to as the Nesbet matrix after the author who first proposed it as being representative of a general configuration interaction matrix [1]. The Nesbet matrix has off-diagonal elements which are unity and diagonal elements which form the sequence of positive odd numbers (i.e. 1, 3, 5,..., $2N + 1$). A second matrix, the "modified Nesbet" matrix, is identical except for replacement of a few of the smallest diagonal elements. For the present tests, the diagonal element values of 3, 5, 7, and 9 were replaced by 1.1, 1.2, 1.3, and 1.4. The set of lowest eigenvalues are closer together than those which are found for the Nesbet matrix itself. The lowest eigenvalues of these two matrices are displayed in Table I for reference purposes.

TABLE I

Lowest Eigenvalues of Test Matrices

	Matrix	
n	Nesbet	Modified Nesbet
1	.29627999	.03360804
2	2.33793249	.14325149
3	4.36505893	.25197477
4	6.38629380	.36234267
5	8.40428470	2.34942119
6	10.42021692	10.34995782
7	12.43473245	12.39437281
8	14.44822251	14.42087465

Initial vectors to form approximations to the eigenvectors were obtained in all cases by extracting a portion of the matrix corresponding to the ten smallest diagonal elements. The full ten by ten submatrix was then fully diagonalized by a standard eigenvalue method. The resulting eigenvectors were then distributed into the arrays for the large vectors and the elements corresponding to rows other than the ten were set to zero. The quantity ten is somewhat of an arbitrary choice, whereas in general this procedure to obtain the necessary starting vectors is a good one. One should always extract initial vectors by using a submatrix whose dimension is larger than the number of vectors being iterated in the SCR (or any other) procedure.

The convergence of eigenvectors is monitored by computing the residual for each major iteration. The residual for the kth vector, R_k, is defined to be

$$R_k = R(x_k) = \| r_k \| \tag{23}$$

where

$$r_k = (A - \lambda_k I)\, x_k / \| x_k \| \tag{24}$$

with

$$\| x_k \| \equiv (x_k{}^T x_k)^{1/2}. \tag{25}$$

The quantity R_k is invariant with respect to the normalization of x_k. As x_k approaches an eigenvector corresponding to the eigenvalue λ_k, each component or r_k approaches zero. (In practice, residuals were computed using the current Rayleigh quotients to approximate the eigenvalues. Since eigenvalues converge faster than the eigenvectors there is assumed to be no visible effect on the general values of R_k.)

Several experiments were made to observe the convergence properties of the SCR algorithm. Each was performed on both the Nesbet and modified-Nesbet matrices. The experiments include: (A) varying the dimension (p) of the SCR vector subspace, (B) a comparison of optimal CR/RS with SCR, and (C) changing the frequency (f) of the generalized eigenproblem substep. The results of these experiments are described in the following sections.

Convergence Versus SCR Subspace

By design, the convergence of a vector to an eigenvector by the SCR algorithm should be improved if its vector subspace converges to a subspace which also contains the eigenvectors whose eigenvalues are adjacent to that one which is sought. The SCR procedure obtains the eigensolutions which correspond to an extreme of the spectrum and so the adjacency property is assured. To observe the change in con-

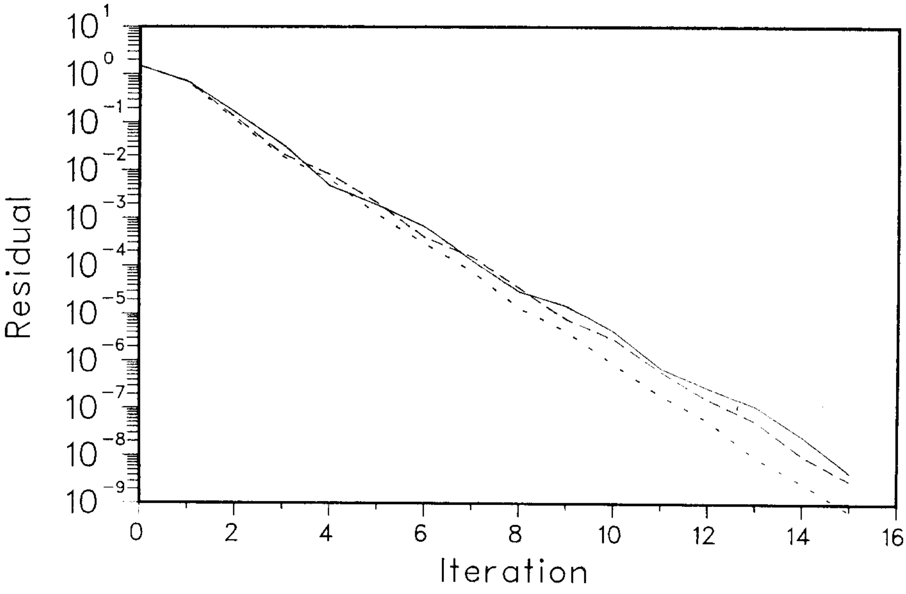

FIG. 1. *SCR* convergence of the lowest eigenvector of the Nesbet matrix: —— $p = 1$, --- $p = 3$, ··· $p = 5$.

vergence rate it is sufficient to fix upon a given eigensolution and increase the subspace size (p); the added vectors converge to some *less* extreme eigenvectors. In Figs. 1 and 2 the convergence of the residual (on a logarithmic scale) is plotted versus the major iteration number for the *lowest* eigenvector of the Nesbet and modified-Nesbet matrices respectively. In general it is observed that convergence is better when the subspace is larger. The reason for the diminished convergence rates seen in Fig. 2 is the nature of the modified-Nesbet matrix. The extreme eigenvalues are more closely spaced and the corresponding eigenvectors do not have a single, clearly-dominant component. Convergence is best when the subspace contains all vectors which will converge to eigenvectors whose eigenvalues are "close" (in the units of this particular matrix) to the one (or ones) being sought. Heuristically, the convergence rate here might be viewed as similar in character to that of the method of simultaneous iteration. For that method it has been shown that the rate is proportional to the ratio of the absolute value of the eigenvalue sought and the nearest absolute eigenvalue not obtained by the simultaneous iteration process. It is seen that when using the SCR method there will be no diminished convergence rate if the chosen subspace is large enough so that all eigensolutions which are close to those which are desired are included. In Fig. 2, the cases $p = 1$, 3, and 5 are a clear demonstration of that effect.

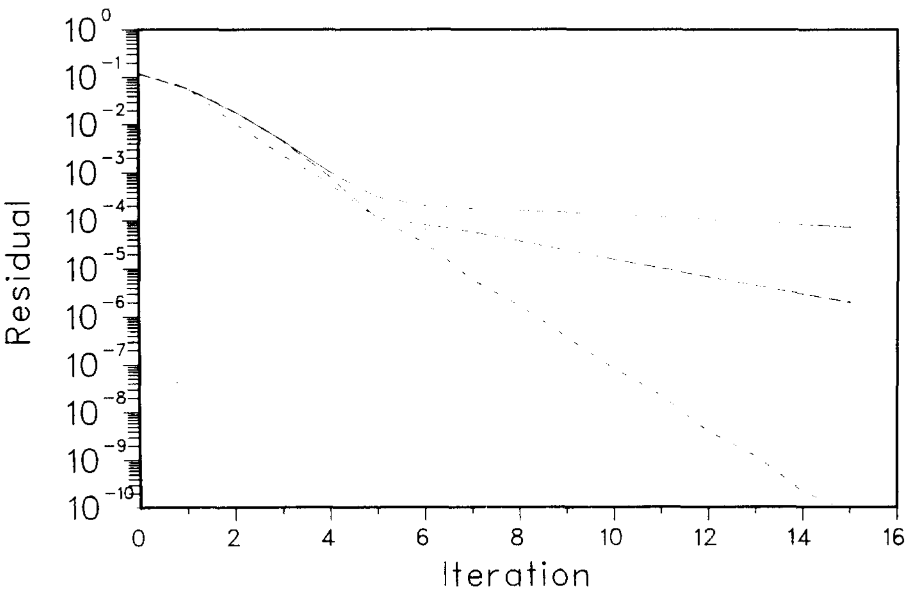

FIG. 2. *SCR* convergence of the lowest eigenvector of the modified Nesbet matrix: —— $p = 1$, – – –$p = 3$, ··· $p = 5$.

Optimal CR/RS Versus SCR

Figures 3 and 4 show a comparison of basic optimal root-shifted coordinate

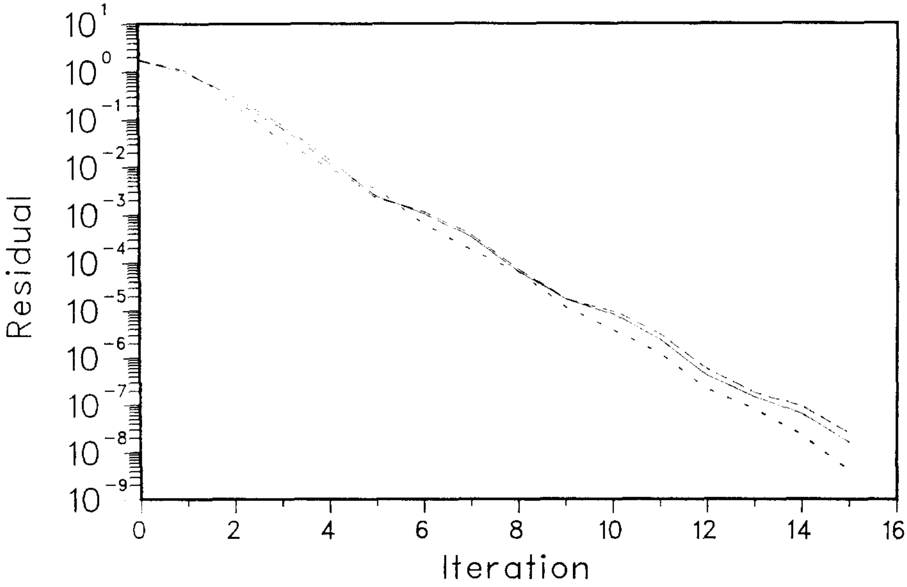

FIG. 3. Optimal and simultaneous *CR* convergence of the third lowest eigenvector of the Nesbet matrix: —— Optimal *CR*, - - - *SCR* ($p = 3$), ··· *SCR* ($p = 5$).

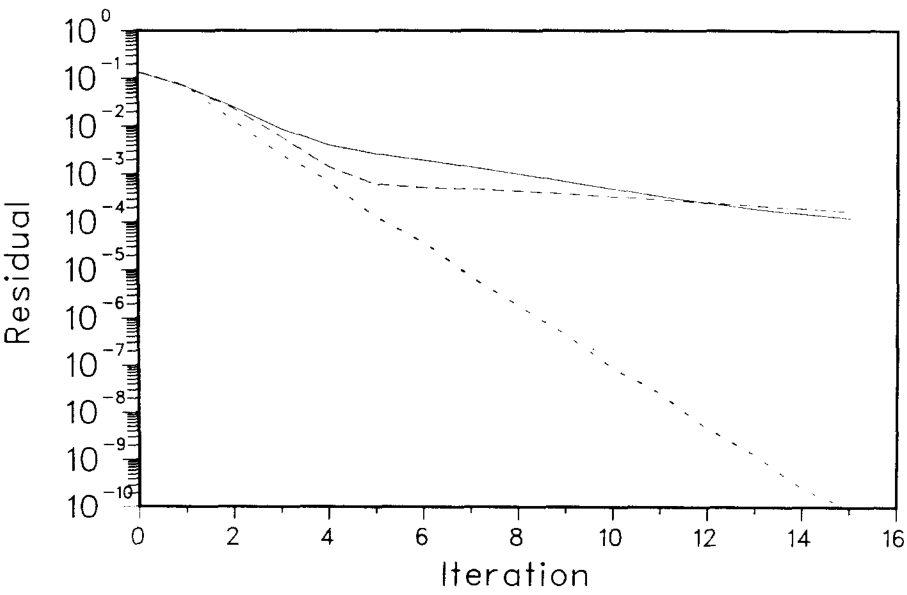

FIG. 4. Optimal and simultaneous *CR* convergence of the third lowest eigenvector of the modified Nesbet matrix: —— Optimal *CR*, - - - *SCR* ($p = 3$), ··· *SCR* ($p = 5$).

relaxation with the present simultaneous coordinate relaxation for the Nesbet and modified-Nesbet matrices respectively. In this case the convergence of the residual for the *third* eigenvector is plotted versus major iteration number. For the optimal CR/RS method, highly accurate first and second eigenvectors were obtained for the root-shifting process. In Fig. 3 no major change in the convergence rates is observed; convergence is rapid.

In Fig. 4 the diminished convergence rate characteristic of close eigenvalues is observed again. Convergence of the optimal CR/RS method is likewise affected as was expected for the SCR method with $p = 3$. There is no way to modify the CR/RS method to correct the deficiency which leads to the diminished convergence. However, if such diminshed convergence is observed when using the SCR method, the dimension of the subspace (p) may be increased as necessary so that improved convergence is obtained.

The Generalized Eigenproblem Substep

To demonstrate the effect of changing the frequency of applying the generalized eigenproblem substep, plots were made which show the convergence of the *second* eigenvector residual when a group of three vectors ($p = 3$) are iterated in the SCR procedure. Figures 5 and 6 show this effect for the Nesbet and modified Nesbet matrices respectively. In Fig. 5 it is clear that the higher the frequency is, the better is the convergence. The gain in convergence speed between a frequency of 1 and 1/2

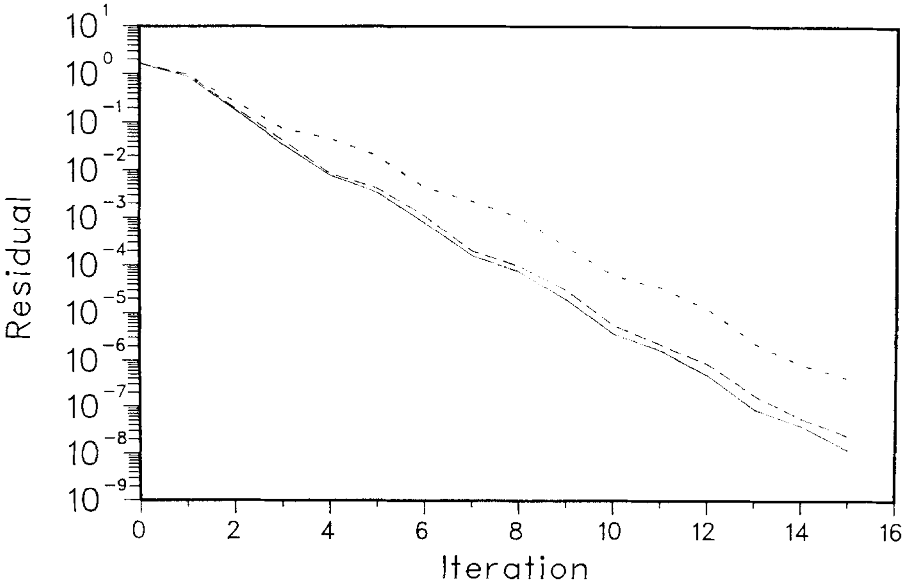

FIG. 5. *SCR* convergence of the second eigenvector of the Nesbet matrix where the frequency of applying the generalized eigenproblem substep is changed: —— $f = 1$, - - - $f = 1/2$, ··· $f = 1/4$.

is not so large for this case, whereas a frequency of 1/4 leads to an undesirable slowing. For the modified Nesbet matrix (Fig. 6) the situation is one of slowed convergence because the iteration subspace is small ($p = 3$). Higher frequencies of application of the substep do show improved convergence but the rates for all of the frequencies shown are virtually unchanged. Other tests which cannot be documented here in detail show that higher frequency of application of the substep improves convergence.

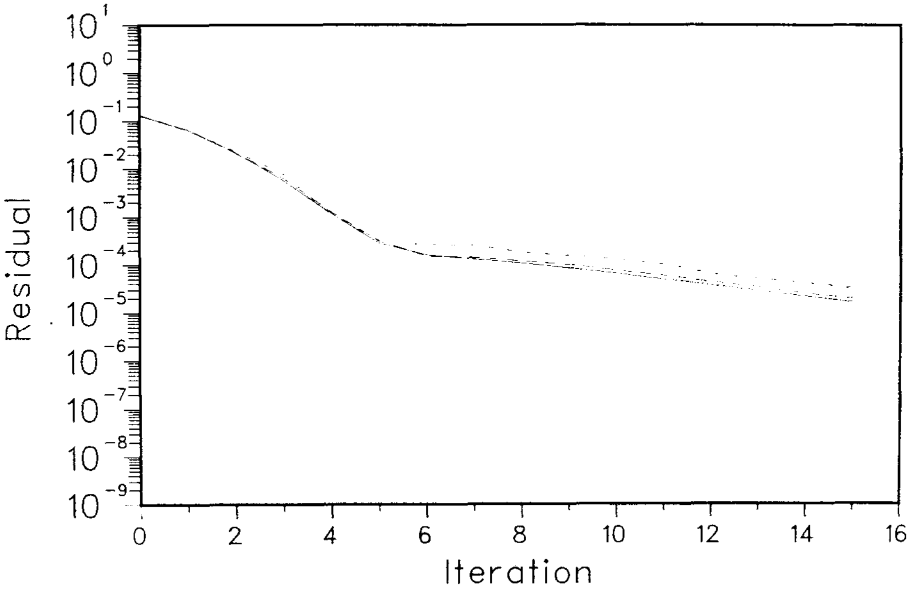

FIG. 6. *SCR* convergence of the second eigenvector of the modified Nesbet matrix where the frequency of applying the generalized eigenproblem substep is changed: —— $f = 1$, - - - $f = 1/2$, ··· $f = 1/4$.

This is almost certainly due to having better current vectors and consequently better Rayleigh quotients (after the substep) to use for the following major iteration. It is recommended that if it is worthwhile in terms of total computation time, i.e. when the density of the matrix is small ($d(A) \sim N$), then a frequency of 1/2 should be used to make the algorithm more effective. A good program would test the density and modify the frequency factor as necessary.

COMPARISON WITH OTHER METHODS

In addition to the optimal relaxation methods of Shavitt, et al. [3] another method, that due to Davidson [7], is in wide use for solving the large, sparse matrix eigenvalue problems generated in quantum chemical configuration interaction calculations. That method has the considerable advantage of not requiring the matrix to be accessible by rows because it utilizes only Krylov-like vectors. It solves for eigenvalues and eigenvectors one at a time and the algorithm is designed and organized for a computer environment consisting of a relatively small central memory and essentially no charge for the large amount of *I/O* processing required. The present algorithm is designed to be run on a computer system for which *I/O* charges are a considerable fraction of the total expanse and the central memory is not a limiting factor. The two methods are complementary and from all indications are quite comparable in their use of central processor time. It is expected that neither algorithm will consistently win out over the other in terms of the total number of matrix vector products, but that each will perform best on some particular type of matrix. Only as more detailed experience with both methods is accumulated will the nature of any specific problem dependence become well-defined.

A generalization of Davidson's expansion method due to B. Liu [8] has also been described. That algorithm is a simultaneous-vector version which is organized so that *I/O* processing is more efficient. It improves the convergence properties of the single vector algorithm [7] and also retains the advantage of not requiring an ordered matrix.

Summary

An algorithm is proposed for a scheme of simultaneous coordinate relaxation. A variant of root-shifting coordinate relaxation, this procedure instead consists of iterating several vectors *at the same time*, instead of one at a time. For many matrix eigenvalue problems for which coordinate relaxation is a viable procedure, the present algorithm will be more effective than previous implementations of coordinate relaxation. Total central processor operations should be decreased due to significantly improved convergence, and there should be less non-arithmetic overhead charged for peripheral operations because there is more work to be done with the external information each time it is brought into central memory.

Acknowledgment

This paper was prepared as a result of work performed in part under NASA Contract NAS1-14101 while the author was at ICASE, NASA Langley Research Center.

References

1. R. K. Nesbet, *J. Chem. Phys.* **43** (1965), 311.
2. I. Shavitt, *J. Computational Phys.* **6** (1970), 124.
3. I. Shavitt, C. F. Bender, A. Pipano, and R. P. Hosteny, *J. Computational Phys.* **11** (1973), 90.

4. F. L. Bauer, *Z. Angew. Math. Phys.* **8** (1957), 214. [See also H. Rutishauser, *Numer. Math.* **13** (1969), 4.]
5. H. R. Schwarz, *Comput. Methods Appl. Mech. Eng.* **3** (1974), 11.
6. A. Ruhe, *Math. Comput.* **28** (1974), 695.
7. E. R. Davidson, *J. Computational Phys.* **17** (1975), 87.
8. B. Liu, private communication (1978).