The Iterative Calculation of Several of the Lowest or Highest Eigenvalues and Corresponding Eigenvectors of Very Large Symmetric Matrices

I. SHAVITT

Battelle Memorial Institute, Columbus, Ohio 43201
and Department of Chemistry, Ohio State University, Columbus, Ohio 43210

AND

C. F. BENDER*, A. PIPANO†, AND R. P. HOSTENY‡,

Battelle Memorial Institute, Columbus, Ohio 43201

Received February 23, 1972

The coordinate relaxation method for the iterative calculation of the lowest (or highest) root of a symmetric matrix, based on the minimization (or maximization) of the Rayleigh quotient, has been generalized to make it possible to obtain several of the lowest (or highest) roots in order without explicitly modifying the original matrix. The method is particularly suitable for very large matrices (even of order 10^4 or more), especially if they are sparse, such as those which occur in large-scale configuration interaction calculations. A modified δ^2 extrapolation procedure has been found to accelerate convergence in the more difficult cases, such as those involving nearly degenerate roots.

I. INTRODUCTION

Large-scale configuration interaction (CI) calculations of electronic wavefunctions of atoms and molecules have become practical and increasingly common in recent years [1–4]. Large symmetric matrices, typically of order a few hundred or a few thousand, are produced in such calculations, and one or more of their lowest eigenvalues and corresponding eigenvectors are required. These

* Present address: Lawrence Radiation Laboratory, University of California, Livermore, California 94550.

† Present address: Israel Aircraft Industries, Ltd., Lod Airport, Israel.

‡ Present address: Chemistry Division, Argonne National Laboratory, Argonne, Illinois 60439.

matrices usually have a dominant main diagonal and are sparse (sometimes no more than 5 % of their elements are nonzero [4], but these elements are generally distributed in an irregular manner throughout the matrix), so that iterative methods, particularly those which can be made to use and preserve the sparseness of the matrix throughout the calculation, are most suitable for the determination of the eigenvalues and eigenvectors. The size of the matrices is often such that they cannot be accommodated completely in the computer's central memory, and therefore the most appropriate eigenvalue procedures are those which can easily be organized to require small sections of the matrix at a time in some convenient order (such as using one row at a time of the lower triangle of the matrix, sequentially, preferably with zero elements omitted [5, 6]). It is also convenient if the original matrix is not modified in the course of the calculation, so that repeated rewriting of the modified matrix on auxiliary storage is not required.

A method which has frequently been used for extracting the lowest eigenvalue of large CI matrices is that of Nesbet [7], which can easily be organized to satisfy all the desiderata discussed above [5, 6]. It is an iterative procedure, essentially identical to Cooper's relaxation method [8], in which one component at a time of a trial vector is adjusted so as to satisfy the corresponding equation of the eigenvalue problem (using the Rayleigh quotient of the trial vector as the current eigenvalue estimate), except that Nesbet's algorithm also provides a very convenient formula for updating the Rayleigh quotient after each adjustment.

An alternative iterative approach for finding the lowest (or highest) eigenvalue of a symmetric matrix **A** is based on the minimization (or maximization) of the Rayleigh quotient

$$\rho(\mathbf{v}) = (\mathbf{v}, \mathbf{A}\mathbf{v})/(\mathbf{v}, \mathbf{v}) \tag{1}$$

with respect to the trial vector **v**. One procedure using this approach is the gradient method of Hestenes and Karush [9, 10], in which each iteration consists of adjusting **v** by adding to it an appropriate multiple of the gradient of ρ,

$$\nabla\rho(\mathbf{v}) = 2[\mathbf{A} - \rho(\mathbf{v})]\mathbf{v}/(\mathbf{v}, \mathbf{v}). \tag{2}$$

Alternatively, this approach can be cast in a relaxational form (see, for example, Fadeev and Fadeeva [11], who refer to this as the "method of coordinate relaxation"); in this case only one component of **v** is adjusted in each step so as to decrease (or increase) $\rho(\mathbf{v})$ as much as possible, a complete iteration consisting of a cycle of such adjustments of all components in turn, and the iterations are continued until all the adjustments in a complete cycle are sufficiently small. The optimum adjustment in each step is obtained very easily by the solution of a simple quadratic equation, and the Rayleigh quotient of the adjusted vector is also easily obtained without referring back to the matrix **A**. This procedure (which we shall refer to as the method of optimal relaxations, or MOR) is the basis for the techniques proposed in the present paper for computing several eigenvalues and eigenvectors of large matrices and has been used successfully at this laboratory for matrices of order up to ~8000. A variant of MOR in which several components of $\mathbf{v}$ are optimized simultaneously, requiring a "mini-iteration" in each step (involving the repeated solution of a small set of linear simultaneous equations) in order to find the optimal adjustments, has been used in CI calculations by Bender and Davidson [2, 12]. A variant of the gradient method which also minimizes the Rayleigh quotient relative to a small subspace of $\mathbf{A}$ at a time has been proposed by Karush [13] (see also Hestenes [14]). An extension of the gradient procedure to include the effect of higher derivatives has been developed by Empedocles [15].

Examination of higher roots has been quite uncommon in the case of very large CI matrices, partly because the matrix is often derived from a basis set which is only appropriate for the description of the lowest eigenstate, but also because the above methods, in their basic form, are unsuitable for the determination of any but the lowest (and highest) root. Actually, a practical procedure for obtaining additional eigenvectors with the gradient method by an approach due to Lanczos [16] has been pointed out by Hestenes and Karush [9, 10], but, as we shall see later, the gradient method is less convenient for very large matrices than the relaxation methods. Cooper [8] has shown a very convenient technique for the computation of additional eigenvectors in a somewhat specialized situation, and a generalization of his approach forms the basis for one of the two methods proposed in the present paper.

Experience at this and other laboratories has shown that the basic form of the Cooper–Nesbet method (as well as the method of optimal relaxations) is very effective for the determination of the lowest eigenvalue of large CI matrices,[1] but it is generally almost impossible to get it to converge to a higher root, even with a reasonably good starting approximation, particularly if this root is within the range of values spanned by the diagonal elements of the matrix (for a brief discussion of the convergence properties of Nesbet's method, see [5]). It has been found in this laboratory that repeated orthogonalization of the trial vector to the previously obtained lower eigenvectors throughout the iterations (suggested, for example, by Hestenes and Karush [9]) could often overcome this problem, and an efficient procedure for carrying out the necessary Gram–Schmidt orthogonalizations without excessive recomputations has been devised by Gilman [17],

[1] In the original form of this method as proposed by Nesbet [7] one component of the trial vector (usually the largest) is kept constant throughout the iterations; this has however been found to make convergence very slow when the corresponding component of the normalized eigenvector was smaller than about 0.8. Subsequent experience has shown that allowing *all* components to participate in the relaxation process corrects this deficiency, and convergence is then generally good even if no component of the eigenvector is dominant.

but even then the method was sometimes ineffective and often slow. Traditional deflation methods (see, e.g., Wilkinson [18]) could in principle be used, after each root is found, in order to remove that eigenvalue and the corresponding eigenvector from the matrix (reducing its order by 1) so that the next root can then be determined by the Cooper–Nesbet or optimal relaxations method. But, if applied explicitly, this means that the original matrix must be modified extensively (and quite laboriously), destroying its sparseness in the process.

Well-known methods of matrix diagonalization, such as those of Givens and of Householder (see, e.g., Wilkinson [18]) cannot be easily applied to matrices which are too big to fit into the computer's central memory and cannot be conveniently made to take advantage of the matrix's sparseness,[2] thus they have generally been used in relatively small CI calculations only. Furthermore, the tridiagonalization step in these methods involves a computational effort which is proportional to n^3 (where n is the order of the matrix), while the relaxation methods require an effort which is approximately proportional to n^2 (for each root sought). In the present paper it will be shown that the method of optimal relaxations can easily be generalized to provide a quite simple procedure for obtaining several of the lowest (or highest) roots, in order, of large sparse matrices, without sacrificing any of the features which make it so convenient and effective for the single lowest (or highest) root.

Section II establishes the notation, discusses the basic form of the method of optimal relaxations, and examines its relationship to the Cooper–Nesbet and gradient methods. Sections III and IV describe two alternative generalizations for the determination of several roots. Both these approaches have been found to be quite effective in a number of CI calculations involving thousands of configurations each [4, 22], and no definite preference for one or the other has been established by the authors. An extrapolation procedure to speed up convergence in problematic cases is described in Section V.

II. The Method of Optimal Relaxations for the Lowest Root of a Symmetric Matrix

1. *Notation*

Given a real symmetric matrix $\mathbf{A}$ of order n, its eigenvalues and eigenvectors will be denoted by λ_i and $\mathbf{x}_i$, respectively,

$$\mathbf{A}\mathbf{x}_i = \lambda_i \mathbf{x}_i \qquad (i = 1, 2, \ldots, n), \tag{3}$$

[2] For methods which attempt to overcome this problem see Tewarson [19]. A review of recent work on computations with sparse matrices has been given by Tewarson [20]; a number of pertinent papers have appeared in the proceedings of a recent conference [21].

with the λ_i arranged in increasing order,

$$\lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_n , \tag{4}$$

and the eigenvectors chosen orthonormal,

$$(\mathbf{x}_i , \mathbf{x}_j) = \delta_{ij} \qquad (i, j = 1, 2,..., n). \tag{5}$$

[While the analysis in this paper deals with real matrices and vectors only, the generalization to the case in which **A** is complex hermitian is straightforward; similarly, while the treatment is for the ordinary eigenvalue problem (3), the generalized problem [5, 7, 10, 23],

$$\mathbf{A}\mathbf{x}_i = \lambda_i \mathbf{S}\mathbf{x}_i , \tag{6}$$

where **S** is a given, symmetric, positive definite matrix, can easily be dealt with by replacing every scalar product of the form (**a**, **b**) (not containing **A**) by the corresponding form (**a**, **Sb**), including the scalar products in the denominator of (1) and in the orthonormality expression (5).]

Lower case Roman subscripts will be used to distinguish different eigenvalues and eigenvectors, as in (3), while Greek lower case subscripts will denote components of matrices (as in $A_{\mu\nu}$) and vectors (e.g., v_μ). The μ-th component of the i-th eigenvector will be written as $x_{\mu i}$. The notation $\mathbf{e}_\mu$ will be used for the unit base vector in which only the μ-th component is nonzero,

$$(\mathbf{e}_\mu)_\nu = \delta_{\mu\nu} . \tag{7}$$

If **v** is any vector, we shall define

$$p(\mathbf{v}) = (\mathbf{v}, \mathbf{A}\mathbf{v}), \qquad q(\mathbf{v}) = (\mathbf{v}, \mathbf{v}), \qquad \rho(\mathbf{v}) = p(\mathbf{v})/q(\mathbf{v}), \tag{8}$$

where $\rho(\mathbf{v})$ is of course the Rayleigh quotient of **v**.

Everything which will be said here of the lowest eigenvalues will also apply, of course, with suitable modifications, to the highest eigenvalues.

2. *Properties of the Rayleigh Quotient* [9]

Let **v** be an arbitrary vector, and let its decomposition in terms of the eigenvectors of **A** be written in the form

$$\mathbf{v} = \sum_{i=1}^{n} c_i(\mathbf{v})\, \mathbf{x}_i . \tag{9}$$

The Rayleigh quotient (8) is expressible as a weighted average of the eigenvalues,

$$\rho(\mathbf{v}) = \sum_i \lambda_i c_i^2(\mathbf{v}) \Big/ \sum_i c_i^2(\mathbf{v}). \tag{10}$$

It is a homogeneous function of $\mathbf{v}$,

$$\rho(k\mathbf{v}) = \rho(\mathbf{v}), \tag{11}$$

and, in particular,

$$\rho(k\mathbf{x}_i) = \lambda_i , \tag{12}$$

where k is an arbitrary constant. It is easily seen that $\rho(\mathbf{v})$ assumes its lowest (or highest) value λ_1 (or λ_n) when $\mathbf{v} = k\mathbf{x}_1$ (or $\mathbf{v} = k\mathbf{x}_n$); in case λ_1 is degenerate, $\lambda_1 = \lambda_2 = \cdots = \lambda_r$ $(r < n)$, $\rho(\mathbf{v})$ will assume its lowest value when $\mathbf{v}$ is any vector in the space of $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_r\}$ (with a similar result for a degenerate λ_n). On the other hand, $\rho(\mathbf{v})$ has a saddle-point whenever $\mathbf{v} = k\mathbf{x}_i$ $(\lambda_1 < \lambda_i < \lambda_n)$.

It is thus clear that the problem of finding the lowest (highest) eigenvalue of $\mathbf{A}$ and a corresponding eigenvector is equivalent to the problem of finding a vector $\mathbf{v}$ for which $\rho(\mathbf{v})$ is a minimum (maximum).

3. *Minimization of* $\rho(\mathbf{v})$ [11]

Given a trial vector $\mathbf{v}$, let us choose a correction vector $\mathbf{w}$, and define a new vector $\mathbf{v}'$ by

$$\mathbf{v}' = \mathbf{v} + \alpha\mathbf{w}. \tag{13}$$

The optimum vector of the form $\mathbf{v}'$ will be obtained by minimizing $\rho(\mathbf{v}')$ with respect to α, after which a new correction vector $\mathbf{w}$ will be chosen and the process repeated. In the gradient method $\mathbf{w}$ is chosen as the gradient of $\rho(\mathbf{v})$ (2), while in the optimal relaxations method $\mathbf{w}$ is taken successively as the different base vectors $\mathbf{e}_\mu$.

For the sake of compactness we shall use the notation

$$\rho = \rho(\mathbf{v}), \qquad \rho' = \rho(\mathbf{v}'), \tag{14}$$

with similar definitions for p, p', q, and q'. Obviously,

$$\begin{aligned} p' &= p + 2\alpha(\mathbf{v}, \mathbf{A}\mathbf{w}) + \alpha^2(\mathbf{w}, \mathbf{A}\mathbf{w}), \\ q' &= q + 2\alpha(\mathbf{v}, \mathbf{w}) + \alpha^2(\mathbf{w}, \mathbf{w}). \end{aligned} \tag{15}$$

As pointed out in [11], the derivative of ρ' with respect to α is given by

$$\begin{aligned} d\rho'/d\alpha = 2\{&[(\mathbf{v}, \mathbf{A}\mathbf{w})q - (\mathbf{v}, \mathbf{w})\,p] + [(\mathbf{w}, \mathbf{A}\mathbf{w})q - (\mathbf{w}, \mathbf{w})\,p]\alpha \\ &+ [(\mathbf{v}, \mathbf{w})(\mathbf{w}, \mathbf{A}\mathbf{w}) - (\mathbf{w}, \mathbf{w})(\mathbf{v}, \mathbf{A}\mathbf{w})]\,\alpha^2\}/\{q + 2(\mathbf{v}, \mathbf{w})\alpha + (\mathbf{w}, \mathbf{w})\,\alpha^2\}^2. \end{aligned} \tag{16}$$

Normally this derivative will vanish at two real finite values of α, and will also tend to zero as $\alpha \rightarrow \pm\infty$; ρ' will then have one maximum and one minimum, and will tend to a constant value $\rho' \rightarrow \rho(\mathbf{w})$ as $\alpha \rightarrow \pm\infty$. Special cases will be considered below (Section II.4). The two extrema will be solutions of the quadratic equation

$$a\alpha^2 + b\alpha + c = 0, \tag{17}$$

where

$$\begin{aligned} a &= [(\mathbf{v}, \mathbf{w})(\mathbf{w}, \mathbf{Aw}) - (\mathbf{w}, \mathbf{w})(\mathbf{v}, \mathbf{Aw})]/q, \\ b &= (\mathbf{w}, \mathbf{Aw}) - (\mathbf{w}, \mathbf{w})\rho, \\ c &= (\mathbf{v}, \mathbf{Aw}) - (\mathbf{v}, \mathbf{w})\rho. \end{aligned} \tag{18}$$

The expressions (18) take on particularly simple forms when $\mathbf{w}$ is chosen as one of the base vectors $\mathbf{e}_\mu$:

$$\begin{aligned} a &= (A_{\mu\mu}v_\mu - f_\mu)/q, \\ b &= A_{\mu\mu} - \rho, \\ c &= f_\mu - \rho v_\mu , \end{aligned} \tag{19}$$

where

$$f_\mu = \sum_{\nu=1}^{n} A_{\mu\nu}v_\nu , \tag{20}$$

and the second Eq. (15) takes the form

$$q' = q + (2v_\mu + \alpha)\alpha. \tag{21}$$

The solution of (17) corresponding to a minimum of ρ' is given by

$$\begin{aligned} \alpha &= \{-b + \sqrt{b^2 - 4ac}\}/(2a) \\ &= 2c/\{-b - \sqrt{b^2 - 4ac}\}, \end{aligned} \tag{22}$$

with the second form more convenient computationally when $b > 0$ (in order to avoid the loss of significant figures as $\mathbf{v} \rightarrow \mathbf{x}_1$ and $c \rightarrow 0$). The Rayleigh quotient of $\mathbf{v}'$ (for any value of α) is easily found from

$$\rho' = \rho + (b\alpha + 2c)\alpha/q', \tag{23}$$

with q' given by (15). When α is given by (22) we find that (23) is equivalent to

$$\rho' = \rho - (\alpha^2/q')(b^2 - 4ac)^{1/2}, \tag{24}$$

confirming that the solution (22) corresponds to a minimization of ρ', while a maximization can be achieved by changing the sign of the radical in (22), and therefore in (24) (as will be shown below, $b^2 \geqslant 4ac$ always, and thus α is always real).

4. *Special Cases*

In order to examine special cases in the solution of (17), let us decompose $\mathbf{w}$ into a vector parallel to $\mathbf{v}$ and another perpendicular to it:

$$\begin{aligned} \mathbf{w} &= \mathbf{u} + \beta\mathbf{v}, \qquad (\mathbf{u}, \mathbf{v}) = 0, \\ \beta &= (\mathbf{v}, \mathbf{w})/q \end{aligned} \tag{25}$$

(in the gradient method we have $\beta = 0$). It is then seen that

$$\begin{aligned} c &= (\mathbf{u}, \mathbf{A}\mathbf{v}), \\ b &= (\mathbf{u}, \mathbf{A}\mathbf{u}) - (\mathbf{u}, \mathbf{u})\rho + 2\beta c, \\ a &= \beta(b - \beta c) - (\mathbf{u}, \mathbf{u})c/q, \end{aligned} \tag{26}$$

so that

$$b^2 - 4ac = [(\mathbf{u}, \mathbf{A}\mathbf{u}) - (\mathbf{u}, \mathbf{u})\rho]^2 + 4(\mathbf{u}, \mathbf{u})\, c^2/q \geqslant 0, \tag{27}$$

confirming that α is always real. Furthermore, the second equality in (27) can only hold if $a = b = c = 0$, in which case ρ' is independent of α (so that we may choose $\alpha = 0$).

Next we consider the case of $a = 0$, but $b \neq 0$; Eq. (17) then has one solution only, $\alpha = - c/b$, which corresponds to a minimum of ρ' if $b > 0$ and to a maximum if $b < 0$. An examination of (16) shows, however, that in the latter case ρ' will have its lowest value, $\rho' = \rho(\mathbf{w})$, at $\alpha \to \pm\infty$, so that the appropriate solution for $b < 0$ is $\mathbf{v}' = \mathbf{w}$. If, on the other hand, $b = 0$, then either a and c are both zero, or they both are nonzero and of opposite sign, in which case the solution corresponding to a minimum of ρ' is $\alpha = \operatorname{sgn}(a)(-c/a)^{1/2}$. Lastly, if $c = 0$, which happens, for example, if either $\mathbf{v}$ or $\mathbf{u}$ is an eigenvector of $\mathbf{A}$, the appropriate solution is $\alpha = 0$ if $b > 0$ and $\alpha = -1/\beta$ if $b < 0$. In this last case $\mathbf{v}' = -\mathbf{u}/\beta$, and it may be more convenient (particularly if β is very small or zero) to use $\mathbf{v}' = \mathbf{u}$.

5. *The Computational Procedure for Optimal Relaxations*

As seen in (19), the calculation of the coefficients of the quadratic equation for one step of MOR is quite straightforward, most of the work going into the calculation of the sum f_μ of (20). This is the same sum that is required in the Cooper–Nesbet relaxational method [5–8]; the extra work involved in MOR in obtaining the solution (22) to the quadratic equation is trivial compared to the calculation of f_μ in the case of large matrices. The same procedures which have been devised [5, 6] for the Cooper–Nesbet algorithm for the calculation of the f_μ sums in the case of very large sparse matrices arranged by rows of the lower triangle, involving the calculation and updating of the partial sums

$$t_\mu = \sum_{\nu=\mu+1}^{n} A_{\nu\mu} v_\nu \,, \tag{28}$$

can be used without change in MOR. This means that f_μ is obtained from the nonzero elements of the μ-th row of the lower triangle only,

$$f_\mu = t_\mu + \sum_{\nu=1}^{\mu} A_{\mu\nu} v_\nu \,. \tag{29}$$

Then a, b, and c are obtained from (19), the increment α is obtained from (22) (or the appropriate special case formula if any of the coefficients is zero or very small), q and ρ are updated by (21) and (23), respectively, v_μ is adjusted by

$$v_\mu{}' = v_\mu + \alpha, \tag{30}$$

and the partial sums t_ν are updated by

$$t_\nu{}' = t_\nu + A_{\mu\nu}\alpha \qquad (\nu = 1, 2,..., \mu - 1) \tag{31}$$

in preparation for the next iteration, using the same matrix elements as in (29). [In the case of the generalized eigenvalue problem (6) it would be necessary to replace (21) by

$$q' = q + (2g_\mu + S_{\mu\mu}\alpha)\alpha, \tag{32}$$

where the sum

$$g_\mu = \sum_{\nu=1}^{n} S_{\mu\nu} v_\nu \tag{33}$$

is computed through equations analogous to (29) and (31).] If α in a given step turns out to be very small we may decide to forgo the adjustment of v_μ at this time and skip any updating of q, ρ, and most importantly, the t_ν. This procedure is carried out for all $\mu = 1, 2,..., n$, in turn; we refer to one such cycle of n steps as an iteration. The process is iterated until all α's for a complete iteration are sufficiently small, and then the last $\mathbf{v}$ is normalized. (In hand computations on small matrices it may be advantageous to use an irregular, judiciously chosen sequence of steps [8], but for computer programs designed for large matrices this would generally be impractical and the ordered sequence $\mu = 1, 2, \ldots, n$ should be used.)

6. *Comparison with Other Methods*

To compare MOR with the Cooper–Nesbet relaxation procedure [7, 8], we apply the binomial expansion to the radical in the first line of (22) and obtain (assuming $b > 0$):

$$\alpha = -(c/b)[1 + (2ac/b^2) + \cdots]. \tag{34}$$

The adjustment used in the Cooper–Nesbet method is $\alpha = -c/b$, which is seen to be the first term of (34). As the solution for the lowest root λ_1 is approached (in which case $b > 0$), we have $c \to 0$ and the two methods become equivalent, but they are definitely inequivalent when ρ is not yet close to λ_1. In particular, when $\mathbf{v}$ is proportional to an eigenvector $\mathbf{x}_i$ ($\lambda_i > \lambda_1$), the Cooper–Nesbet adjustments are all zero and no further change in $\mathbf{v}$ occurs, but in MOR ρ will continue to converge downwards provided $b < 0$ (i.e., $A_{\mu\mu} < \rho$) for at least one value of μ. Of course, the Cooper–Nesbet algorithm may easily be modified, when λ_1 is the desired root, to replace $\mathbf{v}$ by $\mathbf{e}_\mu$ [or by $\mathbf{e}_\mu - (v_\mu/q)\mathbf{v}$] whenever $b < 0$; in fact if the initial trial vector is chosen as $\mathbf{e}_\mu$, where $A_{\mu\mu}$ is the lowest diagonal element of $\mathbf{A}$, then $b \geqslant 0$ automatically (and increases monotonically for each component of $\mathbf{v}$). Both methods cannot guarantee convergence to the lowest root in those cases in which the next higher root is also lower than the lowest diagonal element of $\mathbf{A}$, though it is believed (without proof) that MOR is less likely to fail to reach the lowest root than the other method. Experience has shown that MOR convergence is considerably faster than the Cooper–Nesbet procedure when the initial trial vector is a very poor estimate for $\mathbf{x}_1$, but, as expected, the rates of convergence are comparable when reasonably good trial vectors are obtainable.

Comparing MOR with the gradient procedures [9, 10], we note that in the latter each iteration, which involves the use of all the (nonzero) elements of $\mathbf{A}$ and $O(n^2)$ multiplications, is about comparable in computational effort (in the case of very large matrices) to a complete iteration, made up of n adjustment steps, of the former. If we denote by i_g and i_r the number of complete iterations required for convergence for the same matrix by the gradient and optimal

in MOR, then our experience with large matrices can be summarized in the inequalities

$$i_r \ll i_g \ll n i_r .$$

In fact, while we find that $i_g = O(n^s)$, where s is not very much less than 1, resulting in a total computational effort for the gradient method of $O(n^{2+s}) \approx O(n^3)$, in contrast i_r appears to be almost independent of n, being of the order of 10 in a very large number of computational experiments with matrices of order up to $\sim 10^4$, leading to a total computational effort for one root by MOR of $O(10n^2)$, which is much more favorable when $n \gg 10$. While the gradient method can be made to yield more than one root in one iterative sequence [9, 10], it is still much slower in the calculation of even a few roots than the generalizations of MOR described in Sections III and IV.

The computational effort in most noniterative methods [18], as well as in some iterative procedures such as the Jacobi and power methods, is generally of $O(n^3)$, and they generally involve considerably increased computation times if the matrix A cannot be fully accommodated in the computer's central memory.

III. The Root-Shifting Method for Higher Eigenvalues

1. *Description of the Method*

The first approach which will be considered for the calculation of additional roots by MOR is a generalization of a procedure applied by Hotelling [24] (see also [18]) to the power method and used by Cooper [8] with his relaxation method for the case of matrices with all positive eigenvalues. The generalization depends on the fact that the matrix

$$\mathbf{A}^{(k)} = \mathbf{A} + \sum_{i=1}^{k-1} q_i \mathbf{x}_i \mathbf{x}_i^T \qquad (k \leqslant n), \tag{35}$$

where $\mathbf{x}_i$ are orthonormal eigenvectors of $\mathbf{A}$ and q_i are arbitrary constants, has the same eigenvectors as $\mathbf{A}$

$$\mathbf{A}^{(k)}\mathbf{x}_i = \lambda_i^{(k)}\mathbf{x}_i\,, \tag{36}$$

and its eigenvalues are related to the roots of $\mathbf{A}$ by

$$\begin{aligned} \lambda_i^{(k)} &= \lambda_i + q_i \qquad && (i = 1, 2, \ldots, k-1), \\ \lambda_i^{(k)} &= \lambda_i && (i = k, k+1, \ldots, n). \end{aligned} \tag{37}$$

This is easily verified by direct substitutions. [In the case of the generalized eigenvalue problem (6), where orthonormality is defined with respect to a positive definite symmetric matrix S, the product $\mathbf{x}_i\mathbf{x}_i^T$ in (35) should be replaced by $\mathbf{S}\mathbf{x}_i\mathbf{x}_i^T\mathbf{S}$; this complicates the computational procedure unless S has very few nonzero elements.] Thus, once some eigenvalues λ_i and eigenvectors $\mathbf{x}_i$ $(i = 1, 2, \ldots, k-1)$ of $\mathbf{A}$ have been found, it is possible to "shift" these eigenvalues by a suitable choice of the shift parameters q_i such that the next desired eigenvalue λ_k will be the lowest root of $\mathbf{A}^{(k)}$, to which MOR may be applied. In Hotelling's original formulation the shift parameters were chosen as $q_i = -\lambda_i$, making $\lambda_i^{(k)} = 0$ $(i < k)$; this choice is of course the most appropriate for the power method, and was not too inadequate for Cooper's calculations involving small matrices with all eigenvalues positive. As shall be seen below, more carefully chosen shift parameters are needed for the relaxation methods in the general case with large matrices.

Obviously, while $\mathbf{A}$ may be sparse, $\mathbf{A}^{(k)}$ $(k > 1)$ is generally not sparse at all, and the simple replacement of $\mathbf{A}$ by $\mathbf{A}^{(k)}$ would make this method quite unsuitable in the light of the discussion in Section I. However, $\mathbf{A}^{(k)}$ need never be computed explicitly, since the only quantities involving it in the calculation are the scalar products which appear in (18). It is easily seen that if $\mathbf{a}$ and $\mathbf{b}$ are any two real vectors, then

$$(\mathbf{a}, \mathbf{A}^{(k)}\mathbf{b}) = (\mathbf{a}, \mathbf{A}\mathbf{b}) + \sum_{i=1}^{k-1} q_i(\mathbf{a}, \mathbf{x}_i)(\mathbf{x}_i, \mathbf{b}). \tag{38}$$

(Note that (38), unlike (36-37), does not depend for its accuracy on the $\mathbf{x}_i$'s being exact eigenvectors of $\mathbf{A}$, but only on these vectors being orthonormal.) For the quantities required in (19) we shall have

$$A_{\mu\mu}^{(k)} = (\mathbf{e}_\mu, A^{(k)}\mathbf{e}_\mu) = A_{\mu\mu} + \sum_{i=1}^{k-1} q_i x_{\mu i}^2, \tag{39}$$

$$f_\mu^{(k)} = (\mathbf{e}_\mu, \mathbf{A}^{(k)}\mathbf{v}) = f_\mu + \sum_{i=1}^{k-1} q_i x_{\mu i} r_i, \tag{40}$$

where

$$r_i = (\mathbf{v}, \mathbf{x}_i) \qquad (i = 1, 2, ..., k-1). \tag{41}$$

The scalar products r_i need only be computed once with the initial trial vector $\mathbf{v}$, and are then updated by

$$r_i' = r_i + \alpha x_{\mu i} \qquad (i = 1, 2, ..., k-1), \tag{42}$$

whenever $\mathbf{v}$ is adjusted. The calculation can then proceed in the same manner as before, using the original matrix $\mathbf{A}$, except for the additional short summations of (39–40) and the updating in (42). Of course we have to save the lower eigenvectors $\mathbf{x}_i$ $(i = 1, 2, ..., k-1)$ for the calculation of $\mathbf{x}_k$, but if central memory capacity is a problem these may be stored externally as a rectangular $n \times (k-1)$ matrix $\mathbf{X}$ with components $x_{\mu i}$, arranged by rows, and only the μ-th row brought into central memory for the adjustment step for v_μ; this means however that the matrix $\mathbf{X}$ has to be re-organized on external storage each time a new eigenvector is added. [Again, for the generalized eigenvalue problem (6), $x_{\mu i}$ of (39–42) will have to be replaced by $y_{\mu i} = \sum_\nu S_{\mu\nu} x_{\nu i}$, making the procedure much less attractive unless $\mathbf{S}$ has very few non-zero elements, or unless we store the vectors $\mathbf{y}_i = \mathbf{S}\mathbf{x}_i$.] In cases of slow convergence, in order to reduce the accumulation of rounding errors in the continuous updating of the t_ν's, r_i's, q and ρ, it may be advisable to recompute these quantities occasionally (say after each 10 iterations).

2. *Choice of Shift Parameters*

As each eigenvector $\mathbf{x}_k$ is calculated in turn, the shift parameters q_i needed for the calculation of the next vector have to be redetermined if reasonably rapid convergence is to be achieved. It has been found empirically that large values of these parameters tend to slow down convergence, presumably because they reduce the dominance of the main diagonal in $\mathbf{A}^{(k)}$ (this is why Hotelling's and Cooper's choice of $q_i = -\lambda_i$ is generally inappropriate in the present case). On the other hand the q_i have to be large enough to shift all previously computed eigenvalues well above the next root sought, for convergence is relatively slow if there are any other distinct roots in the near neighborhood of the currently lowest eigenvalue. The following scheme for the choice of the shift constants for the calculation of the k-th root has been found to be near optimum in a large number of test calculations:

(a) Having chosen an initial trial vector $\mathbf{v}_0$ for the k-th root (this choice will be considered further below), orthogonalize it to the previously computed eigenvectors $\mathbf{x}_i$ $(i = 1, 2,..., k - 1)$ by the Schmidt procedure,

$$\mathbf{v} = \mathbf{v}_0 - \sum_{i=1}^{k-1} (\mathbf{x}_i, \mathbf{v}_0)\, \mathbf{x}_i . \tag{43}$$

This ensures that $\rho(\mathbf{v}) \geqslant \lambda_k$.

(b) Compute the quantity

$$q_0 = \rho(\mathbf{v}) - \lambda_1 . \tag{44}$$

(c) Choose the shift constants by the formula

$$q_i = sq_0 - (\lambda_i - \lambda_1) \qquad (i = 1, 2,..., k - 1), \tag{45}$$

where s is a constant somewhat greater than 1; a value of $s = 1.3$ has been found empirically to yield good results in many tests. This procedure results in

$$\lambda_i^{(k)} = \rho(\mathbf{v}) + (s - 1)\, q_0 \geqslant \lambda_k + (s - 1)\, q_0 \qquad (i = 1, 2,..., k - 1). \tag{46}$$

If the initial trial vector was a very poor approximation to $\mathbf{x}_k$ the steps (a)–(c) may be repeated occasionally as $\rho(\mathbf{v})$ is lowered by the iterations.

3. *Convergence*

An analytical study of the convergence of this method has not been carried out, but obviously the considerations which apply for the basic method (Section II) apply here to the matrices $\mathbf{A}^{(k)}$. Again, convergence to the lowest root of $\mathbf{A}^{(k)}$ cannot be guaranteed if the next root λ_{k+1} (or the shifted roots $\lambda_i^{(k)}$) is lower than the lowest diagonal element of $\mathbf{A}^{(k)}$, though such a case of a "skipped" root has not occured in any of our test calculations. Furthermore, a root which has been skipped in the step involving $\mathbf{A}^{(k)}$ is likely to appear in the next step, if the process is continued.

As stated, in most cases the number of iterations required to obtain convergence for each root (to an accuracy of about 10^{-6} for the components of the normalized eigenvector) was of the order of 10 with the matrices obtained in many configuration interaction calculations. Usually only about 7 iterations were needed for the lowest root, and this number increased slowly for higher roots as the denser regions of the spectrum of $\mathbf{A}$ were approached. Convergence was generally very slow for roots which were quite close to the next higher root, even requiring a few hundred iterations in one case involving a pair of very nearly degenerate eigenvalues. An extrapolation method, described in Section V, was found to be very effective in such cases, reducing the number of iterations to a few tens, at most. The case of exact degeneracy was not encountered in any of our calculations, but it seems that no difficulty should occur in that event. (For another treatment of close eigenvalues, in the context of the gradient method, see Rosser *et al.* [25].)

Any inaccuracy in a computed eigenvector would of course greatly reduce the accuracy of the higher eigenvectors computed by the root-shifting method, and if several roots are desired it may be advisable to start out with a smaller convergence criterion for α in the calculations of the first few roots than is required for the higher roots. No problems with error accumulation were encountered with the root-shifting method in our calculations, in which up to 10 roots were calculated for some matrices. Of course, if all or many of the roots are required it would be advantageous to use a different method entirely, such as some version of Householder's method [18, 19, 23], or perhaps the gradient method.

IV. The Orthogonality Constraint Method for Higher Eigenvalues

An alternative approach to the root-shifting method is that of maintaining orthogonality of the trial vector to lower eigenvectors throughout the iterations. Such an approach has been suggested, for example, by Hestenes and Karush [9], and Hotelling [24] has suggested the use of occasional Schmidt orthogonalizations for the removal of lower-vector contaminations (due to rounding errors) when the power method is applied to an initial trial vector which is orthogonal to lower eigenvectors (when the root-shifting procedure is not used).

In the present case the orthogonality constraints can be applied very easily

If $\mathbf{a}$ is any vector, define

$$\mathbf{a}^{(k)} = \mathbf{a} - \sum_{i=1}^{k-1} (\mathbf{x}_i , \mathbf{a})\, \mathbf{x}_i . \tag{47}$$

In particular,

$$\mathbf{e}_\mu^{(k)} = \mathbf{e}_\mu - \sum_{i=1}^{k-1} x_{\mu i}\mathbf{x}_i . \tag{48}$$

If the trial vector $\mathbf{v}$ for the calculation of the k-th root is first replaced by $\mathbf{v}^{(k)}$ according to (47), and if the correction vectors $\mathbf{w}$ of (13) are taken as the $\mathbf{e}_\mu^{(k)}$ of (48) instead of $\mathbf{e}_\mu$, then except for rounding errors the trial vector will remain orthogonal to the lower eigenvectors throughout the iterations. Any contaminations due to rounding can be removed by occasional reorthogonalization of the trial vector.

We now have to recompute the expressions for the coefficients (19) of the quadratic equation by substituting $\mathbf{w} = \mathbf{e}_\mu^{(k)}$ in (18). This is facilitated by the result that for any pair of vectors $\mathbf{a}^{(k)}$ and $\mathbf{b}^{(k)}$, defined as in (47),

$$\begin{aligned}(\mathbf{a}^{(k)}, \mathbf{A}\mathbf{b}^{(k)}) &= (\mathbf{a}, \mathbf{A}\mathbf{b}) - \sum_{i=1}^{k-1} (\mathbf{a}, \mathbf{A}\mathbf{x}_i)(\mathbf{x}_i , \mathbf{b}) - \sum_{i=1}^{k-1} (\mathbf{a}, \mathbf{x}_i)(\mathbf{x}_i , \mathbf{A}\mathbf{b}) \\ &\quad + \sum_{i=1}^{k-1} \sum_{j=1}^{k-1} (\mathbf{a}, \mathbf{x}_i)(\mathbf{x}_j , \mathbf{b})(\mathbf{x}_i , \mathbf{A}\mathbf{x}_j) \\ &= (\mathbf{a}, \mathbf{A}\mathbf{b}) - \sum_{i=1}^{k-1} \lambda_i(\mathbf{a}, \mathbf{x}_i)(\mathbf{b}, \mathbf{x}_i).\end{aligned} \tag{49}$$

It is important to notice that while the first equality of (49) is exact, the second equality, unlike (38) of the root-shifting method, depends for its accuracy on the $\mathbf{x}_i$ ($i = 1, 2,..., k - 1$) being exact eigenvectors of $\mathbf{A}$, and its use is likely to cause a greater sensitivity of the orthogonality constraint procedure to errors in lower eigenvectors. We also have

$$(\mathbf{a}^{(k)}, \mathbf{b}^{(k)}) = (\mathbf{a}, \mathbf{b}) - \sum_{i=1}^{k-1} (\mathbf{a}, \mathbf{x}_i)(\mathbf{b}, \mathbf{x}_i); \tag{50}$$

this depends for its accuracy only on the $\mathbf{x}_i$ being orthonormal.

Applying (49, 50) to the scalar products of (18), with $\mathbf{v}$ and $\mathbf{w}$ replaced by $\mathbf{v}^{(k)}$ and $\mathbf{e}_\mu^{(k)}$, respectively, we get equations similar to (19–23) but with $A_{\mu\mu}$, etc., replaced by:

$$A_{\mu\mu}^{(k)} = A_{\mu\mu} - \sum_{i=1}^{k-1} \lambda_i x_{\mu i}^2, \tag{51}$$

$$v_\mu^{(k)} = v_\mu - \sum_{i=1}^{k-1} r_i x_{\mu i}, \tag{52}$$

$$f_\mu^{(k)} = f_\mu - \sum_{i=1}^{k-1} \lambda_i r_i x_{\mu i}, \tag{53}$$

$$q^{(k)} = q - \sum_{i=1}^{k-1} r_i^2, \tag{54}$$

$$p^{(k)} = p - \sum_{i=1}^{k-1} \lambda_i r_i^2, \tag{55}$$

$$\rho^{(k)} = p^{(k)}/q^{(k)}, \tag{56}$$

with r_i given by (41), and with the second equation in (19) replaced by

$$b = A_{\mu\mu}^{(k)} - \rho^{(k)} e_{\mu\mu}^{(k)}, \tag{57}$$

where

$$e_{\mu\mu}^{(k)} = 1 - \sum_{i=1}^{k-1} x_{\mu i}^2 . \tag{58}$$

Note that (54–56) are only used with the initial trial vector, while $q^{(k)}$ and $\rho^{(k)}$ are updated by

$$q^{(k)\prime} = q^{(k)} + (2v_\mu^{(k)} + e_{\mu\mu}^{(k)}\alpha)\alpha \tag{59}$$

and

$$\rho^{(k)\prime} = \rho^{(k)} + (b\alpha + 2c)\alpha/q^{(k)\prime}, \tag{60}$$

respectively. In the computational procedure based on this analysis we store and update $\mathbf{v}$, not $\mathbf{v}^{(k)}$, throughout the iterations, since (30) is much shorter than the procedure for updating $\mathbf{v}^{(k)}$, and we continue to use (28–29), followed by (51–53), (58), the equivalent of (19) (noting (57)), then (22) and the updatings of (30–31) and (59–60); $\mathbf{v}^{(k)}$ is computed by (52) after convergence has been achieved. [As in the case of the root-shifting method, the procedure for the generalized eigenvalue problem (6) involves the use of the vectors $\mathbf{y}_i = \mathbf{S}\mathbf{x}_i$ instead of $\mathbf{x}_i$.]

As noted previously, the orthogonality constraint procedure is rather sensitive to inaccuracies in the lower eigenvectors, primarily because of the use of the second equation in (49), and thus convergence criteria have to be tighter than in the root-shifting version. This sensitivity can be reduced somewhat if **v** is explicitly orthogonalized to the lower eigenvectors initially and then after every few complete iterations, with all updated quantities being recomputed at these times. Another approach is to save the vectors $\mathbf{f}_i = \mathbf{A}\mathbf{x}_i$ and the scalars $(\mathbf{x}_i, \mathbf{A}\mathbf{x}_j) = (\mathbf{x}_i, \mathbf{f}_j)$ $(1 \leqslant j \leqslant i < k)$ and then use the first form of (49); this may still be quite practical if k is not large. It should also be noted that the n different adjustment vectors $\mathbf{e}_\mu^{(k)}$ $(\mu = 1, 2,..., n)$ are not linearly independent if $k > 1$, but this causes no problems. In case any $\mathbf{e}_\mu^{(k)}$ becomes very small in magnitude, such as when $e_{\mu\mu}^{(k)}$ of (58) is less than .01, the corresponding adjustment step is best skipped.

In general, it has been found that the orthogonality constraint method often converges in slightly fewer iterations than the root-shifting procedure, but the former involves a more complicated computer program and takes somewhat longer per iteration. The total computer time for the same problem is usually about equal by the two methods.

V. Extrapolation Procedure

In cases of slow convergence it has been found helpful to use a modified form of the well-known Aitken's δ^2 extrapolation scheme for the eigenvector components. In the usual δ^2 process, if s, $s' = s + a$, and $s'' = s' + a'$ are successive approximations to some quantity $\bar{s}$, we assume that the increments a, a', etc. are in approximate geometric progression, leading to

$$\bar{s} \approx s'' + (a')^2/(a - a'). \tag{61}$$

To apply this in a given iteration of MOR we have to save all the α's in the previous iteration and use them as the increments a in (61) for the corresponding steps in the current iteration, using the new α for a' and v_μ' for s''. It is generally advisable not to replace the normally computed v_μ' components by the extrapolated values until the end of the iteration, in order to maintain the smoothness of the process, saving the extrapolated components meanwhile in the same array which has been used to save the previous increments α (thus requiring storage for only one additional array of length n for the extrapolation process). When replacing **v** by the extrapolated vector it is also convenient to use this opportunity to recompute the quantities q, p, t_v, and r_i directly and thus reduce the accumulation of rounding errors in the updatings. Also, in the orthogonality constraint procedure this is a convenient time to reorthogonalize **v** to the lower eigenvectors.

Due to the limited smoothness of the iteration process, and especially to deal with cases in which the increments α increased temporarily from one iteration to the next, it was necessary to modify the usual δ^2 process of (61) by applying constraints on the magnitude and sign of the extrapolation increment. Backward extrapolation ($|a'| > |a|$) was found to occur particularly in some cases in which, during the iterations, $\rho(\mathbf{v})$ passed through values close to a higher root than the one being sought (this generally happens in cases of near degeneracy). In such cases the normal extrapolation may pull $\rho(\mathbf{v})$ back towards the higher root and thus retard convergence. The following modification of the δ^2 process was found empirically to produce satisfactory results: A constant h in the range 10 to 30 (usually 20) was chosen, and whenever the ratio of the extrapolation increment $(a')^2/(a - a')$ to the last normal increment a' was found to be negative or greater than h, (61) was replaced by

$$\bar{s} = s'' + ha'.$$

Extrapolation after every 10 iterations was found to be near optimum in most cases (if more than 10 iterations were required for convergence).

For the initial trial vectors it has generally been found adequate, for the k-th root, to use $\mathbf{e}_\mu$, where $A_{\mu\mu}$ is the k-th lowest diagonal element of $\mathbf{A}$.

The total computer central storage required by the procedures described in this paper includes $(k + 3)$ arrays of length n to store the vectors $\mathbf{v}$, $\mathbf{t}$, the current row of the lower triangle of $\mathbf{A}$, the increments α saved for the extrapolation procedures, and the previous eigenvectors $\mathbf{x}_i$. If necessary, as previously stated, the eigenvectors $\mathbf{x}_i$ need not be kept in central memory continuously, and only an array of length $k - 1$ for the current (μ-th) components of all the $\mathbf{x}_i$'s needs to be provided, in addition to two more arrays of this length for the r_i and the λ_i ($i = 1, 2, \ldots, k - 1$). Furthermore, only the nonzero elements of the current row of $\mathbf{A}$, suitably identified, need be stored and used in (29) and (31).

Both the root-shifting and orthogonality constraint procedures, with the modified δ^2 extrapolation, have been used at this laboratory in numerous CI calculations involving matrices of order $\sim 10^2$ to 10^4, containing up to 2×10^6 nonzero elements, with between 1 and 10 roots computed in each case, and no failure to converge in a reasonable number of iterations (usually about 10 per root, but sometimes up to about 10^2) has been observed.

Acknowledgments

One of the authors (I.S.) is grateful to Dr. R. V. M. Zahar for very helpful discussions and for bringing Ref. [11] to his attention. Useful discussions with Dr. R. R. Gilman are gratefully acknowledged.

References

1. C. F. Bunge, *Phys. Rev.* **168** (1968), 92.
2. C. F. Bender and E. R. Davidson, *Phys. Rev.* **183** (1969), 23.
3. J. L. Whitten and M. Hackmeyer, *J. Chem. Phys.* **51** (1969), 5584.
4. R. P. Hosteny, R. R. Gilman, T. H. Dunning, Jr., A. Pipano, and I. Shavitt, *Chem. Phys. Lett.* **7** (1970), 325.
5. I. Shavitt, *J. Computational Phys.* **6** (1970), 124.
6. I. Shavitt, "EIGEN," Program 172, Quantum Chemistry Program Exchange, Indiana University, Bloomington, IN, 1970.
7. R. K. Nesbet, *J. Chem. Phys.* **43** (1965), 311.
8. J. L. B. Cooper, *Quart. Appl. Math.* **6** (1948), 179.
9. M. R. Hestenes and W. Karush, *J. Res. Nat. Bur. Stand.* **47** (1951), 45.
10. M. R. Hestenes and W. Karush, *J. Res. Nat. Bur. Stand.* **47** (1951), 471.
11. D. K. Faddeev and V. N. Faddeeva, "Computational Methods of Linear Algebra" (English Translation), Section 61, W. H. Freeman & Co., San Francisco, CA, 1963.
12. C. F. Bender, Ph.D. Thesis, University of Washington, Seattle, WA, 1968.
13. W. Karush, *Pacific J. Math.* **1** (1951), 233.
14. M. R. Hestenes, *in* "Simultaneous Linear Equations and the Determination of Eigenvalues" (L. J. Page and O. Taussky, Eds.), Chapter 12, National Bureau of Standards Applied Mathematics Series, No. 29, U. S. Government Printing Office, Washington, DC, 1953.
15. P. Empedocles, *J. Computational Phys.* **6** (1970), 113.
16. C. Lanczos, *J. Res. Nat. Bur. Stand.* **45** (1950), 255.
17. R. R. Gilman, private communication.
18. J. H. Wilkinson, "The Algebraic Eigenvalue Problem," Oxford University Press, London, 1965.
19. R. P. Tewarson, *Int. J. Comput. Math.* **2** (1970), 247.
20. R. P. Tewarson, *SIAM Rev.* **12** (1970), 527.
21. R. A. Willoughby (Ed), "Proceedings of the Symposium on Sparse Matrices and Their Applications" (held at IBM Watson Research Center, Yorktown Heights, NY, September, 1968), Report RA 1 (# 11707), International Business Machines Corporation, Yorktown Heights, NY, March 12, 1969.
22. R. P. Hosteny, T. H. Dunning, Jr., R. R. Gilman, A. Pipano and I. Shavitt, "The π-Electron States of Butadiene," to be published.
23. G. Peters and J. H. Wilkinson, *SIAM J. Numer. Anal.* **7** (1970), 479.
24. H. Hotelling, *J. Educational Psychol.* **24** (1933), 417.
25. J. B. Rosser, C. Lanczos, M. R. Hestenes, and W. Karush, *J. Res. Nat. Bur. Stand.* **47** (1951), 291.