JOURNAL OF COMPUTATIONAL PHYSICS 23, 404–424 (1977)

A New Approach to the Determination of Several Eigenvectors of a Large Hermitian Matrix

J. A. R. COOPE AND D. W. SABO

Department of Chemistry, University of British Columbia, Vancouver, British Columbia, Canada, V6T 1W5

Received August 31, 1976

To determine n_A eigenvectors of a large hermitian matrix, an initial transformation is made which uncouples an $n_A \times n_A$ block. This transformation is determined by an $[(n - n_A) \times n_A]$-dimensional partitioning matrix f, whose determination is the main computational problem. The eigenvalues of the n_A-dimensional reduced problem are the required exact eigenvalues of the original matrix, and the eigenvectors give n_A exact components of the required eigenvectors of the original matrix. The remaining components are obtained by simple matrix multiplication. Two classes of algorithms are discussed for determining f, both suitable for large matrices. One of these can be regarded as giving generalizations of Nesbet's method for the lowest eigenvalue. When f is known only approximately, then, provided the n_A-dimensional matrices are calculated in an appropriate way, the n_A eigenvalues are upper bounds to the n_A lowest exact eigenvalues, while a first-order error in f gives a second-order error in the eigenvalues. A generalized variance arises naturally, which bounds the variances of the individual eigenvectors, and so defines error limits.

1. INTRODUCTION

A number of iterative techniques have been developed recently for the calculation of several eigenvectors belonging to the lowest eigenvalues of a large real symmetric matrix [1–6]. These techniques have been particularly useful for obtaining electronic wavefunctions for the lower-lying energy levels of atoms and molecules in large-scale configuration interaction calculations [7]. The matrices arising in such calculations typically have diagonal elements arranged in roughly increasing order, with large variation in the diagonal elements compared to individual off-diagonal elements.

In these cases, where the dimensions of the Hamiltonian matrix range from several hundred to tens of thousands, the traditional matrix diagonalization techniques are not practical. With a matrix so large that it must be stored on some auxiliary device, rather than in the central computer memory, only small sections are directly available at one time. Techniques which involve many successive modifications of the original matrix therefore become very inefficient, besides being vulnerable to significant cumulative round-off error. Further, in techniques in which the entire matrix is brought to tridiagonal form before calculation of a single eigenvector, the determination of a

Copyright © 1977 by Academic Press, Inc.
All rights of reproduction in any form reserved.

ISSN 0021-9991

smaller number of eigenvalues and eigenvectors requires nearly as much work as the determination of all of them.

Iterative techniques, on the other hand, can be formulated in a way which minimizes these difficulties. With proper organization, small sections of the matrix can be used sequentially, and the work per iteration can be made proportional to the actual number of eigenvectors being calculated. For matrices of large dimension, the work per iteration is roughly proportional to the square of the dimension of the original matrix, rather than the third power, as in techniques requiring initial tridiagonalization.

Most iterative techniques for the partial diagonalization of large matrices now available are based on the calculation of successive corrections to an approximate starting vector, to obtain a sequence of vectors converging to a single eigenvector. Since these techniques typically use the maximization or minimization of the Rayleigh quotient, with respect to the approximate eigenvector, as the criterion for the calculation of the appropriate corrections [1–4], the single eigenvector obtained usually corresponds to the largest or smallest eigenvalue. To find other eigenvalues and eigenvectors, the same calculation is repeated, but convergence onto previously calculated eigenvectors is prevented, by using one of several techniques [2].

In this paper, the determination of a small number of eigenvectors of a large hermitian matrix is approached from a quite different point of view. We use an eigenvalue-independent partitioning formalism [9], developed in connection with the construction of effective Hamiltonians, to determine an uncoupling operator which defines a mapping from a vector subspace S_A', defined by a desired n_A-dimensional subset of the eigenvectors, to a subspace S_A, of the same dimension, spanned by n_A basis vectors. The desired eigenvectors and eigenvalues are then obtained by solving a matrix eigenvalue problem of the small dimension n_A. The main computational work is the calculation of the uncoupling operator. This method differs from those referred to above in that all the n_A eigenvalues and eigenvectors are effectively determined simultaneously. This uncoupling procedure can be related to both the vanishing of the variance of the matrix with respect to the projection onto the subspace spanned by the desired eigenvectors, and also to the extremizing of the sum of the Rayleigh quotients of the individual approximate eigenvectors.

In principle, this uncoupling can be carried out for any grouping of the eigenvectors of the matrix, and thus, in principle, any subset of the eigenvectors not actually orthogonal to S_A can be calculated without previous determination of any of the other eigenvectors. In practice, it appears that the calculation of the uncoupling operator may converge best when the eigenvectors correspond to the n_A lowest (or highest) eigenvalues. In most computational situations, it is these eigenvectors and eigenvalues of a large matrix which are of interest.

Section 2 outlines the partitioning formalism on which the approach is based, as adapted particularly to matrix eigenvalue problems. A number of properties of approximate solutions quoted here will be discussed in a more general and detailed analysis of the partitioning method to be given elsewhere [10]. Section 3 discusses computational aspects, and outlines two classes of procedures. Section 4 discusses the

generalized eigenvalue problem for a nonorthogonal basis. Section 5 gives some computational results. Detailed descriptions of the algorithms are given in an Appendix.

2. The Partitioning Formalism

We consider the eigenvalue problem

$$Hx_i = \lambda_i x_i , \tag{2.1}$$

for an $n \times n$ hermitian matrix H, in the form

$$X^\dagger H X = \Lambda; \tag{2.2a}$$

$$X^\dagger X = 1_n , \tag{2.2b}$$

where X is the $n \times n$ unitary matrix whose columns are the orthonormal eigenvectors x_i of H, and Λ is the diagonal matrix of the eigenvalues. Let the basis vectors be partitioned into two sets, of dimensions n_A and n_B, defining subspaces of the original underlying n-dimensional vector space, S_A and S_B, so that H takes the partitioned form

$$H = \begin{bmatrix} H_{AA} & H_{AB} \\ H_{BA} & H_{BB} \end{bmatrix}. \tag{2.3}$$

Let the eigenvectors x_i be likewise partitioned into two sets A and B, of dimensions n_A and n_B, defining eigenspaces S_A' and S_B', the eigenvectors in S_A' being the ones of interest. The eigenvector matrix is partitioned columnwise $X = (X^{(A)}, X^{(B)})$ with respect to S_A' and S_B', and row-wise with respect to the basis spaces S_A and S_B, and then takes a block form

$$X = \begin{bmatrix} X_{AA} & X_{AB} \\ X_{BA} & X_{BB} \end{bmatrix}. \tag{2.4}$$

It can be shown, because X is unitary, that it can be factored in the manner

$$X = T\hat{X}, \tag{2.5}$$

where $\hat{X}$ contains only the diagonal blocks of X,

$$\hat{X} = \begin{bmatrix} X_{AA} & 0 \\ 0 & X_{BB} \end{bmatrix}, \tag{2.6}$$

and where

$$T = \begin{bmatrix} 1_A & -f^\dagger \\ f & 1_B \end{bmatrix}, \tag{2.7}$$

with

$$f = X_{BA}X_{AA}^{-1} = -(X_{AB}X_{BB}^{-1})^{\dagger}. \tag{2.8}$$

The $(n_B \times n_A)$-dimensional matrix f may be called the partitioning, or uncoupling, operator. It exists if X_{AA}^{-1} exists, which is the case as long as no vector in S_A' is orthogonal to S_A, that is, as long as none of the A eigenvectors is orthogonal to all the A-basis vectors; and thus f always exists for some partitioning of the basis.

In view of Eq. (2.5), the eigenvalue equations (2.2) can be rewritten

$$\hat{X}^{\dagger}G\hat{X} = \Lambda, \tag{2.9a}$$

$$\hat{X}^{\dagger}g\hat{X} = 1_n, \tag{2.9b}$$

where

$$G = T^{\dagger}HT, \tag{2.10}$$

and

$$g = T^{\dagger}T. \tag{2.11}$$

Because of the form (2.7) of T, the metric g is automatically block diagonal,

$$g = \begin{bmatrix} g_A & 0 \\ 0 & g_B \end{bmatrix}, \tag{2.12}$$

the diagonal blocks being

$$g_A = 1_A + f^{\dagger}f, \tag{2.13a}$$

$$g_B = 1_B + ff^{\dagger}. \tag{2.13b}$$

The diagonal blocks of G are given by

$$G_A = H_{AA} + H_{AB}f + f^{\dagger}H_{BA} + f^{\dagger}H_{BB}f, \tag{2.14a}$$

$$G_B = H_{BB} - H_{BA}f^{\dagger} - fH_{AB} + fH_{AA}f^{\dagger}, \tag{2.14b}$$

and the objective of the calculation is to determine f in such a way that the off-diagonal blocks vanish. The condition for this is

$$D(f) = 0, \tag{2.15}$$

where

$$D(f) = G_{BA} = H_{BA} + H_{BB}f - fH_{AA} - fH_{AB}f. \tag{2.15a}$$

Equation (2.15) can be regarded as a system of simultaneous nonlinear equations for the matrix elements of f. When it is satisfied, Eqs. (2.9) yeild two separate eigen-

value problems, one of dimension n_A for the eigenvalues and eigenvectors of interest,

$$X_{AA}^{\dagger} G_A X_{AA} = \Lambda^{(A)}, \tag{2.16a}$$

$$X_{AA}^{\dagger} g_A X_{AA} = 1_A, \tag{2.16b}$$

and a similar one, of dimension n_B, for the remaining eigenvalues.

The matrices G_A and g_A in (2.16) may equally well be interpreted either as matrices of "effective" operators in the original basis defining S_A, or as matrices of the original operators H and 1_n in a transformed basis spanning S_A'. The language used in the following is that appropriate to the first alternative, Eqs. (2.16) being considered as an eigenvalue problem in the basis space S_A, the two spaces S_A and S_B having been uncoupled.

Equations (2.16) are equivalent to an eigenvalue problem of the form

$$G_A x_{Ai} = \lambda_i g_A x_{Ai}, \tag{2.17}$$

where x_{Ai} is the ith column of X_{AA}, or equivalently to

$$\hat{H}_A x_{Ai} = \lambda_i x_{Ai}, \tag{2.18}$$

where $\hat{H}_A$ is the non-self-adjoint operator

$$\hat{H}_A = g_A^{-1} G_A. \tag{2.19}$$

Equations (2.17) or (2.18) can be solved by any standard technique. Assuming that the quantities g_A and G_A or $\hat{H}_A$, appearing in (2.17) and (2.18) are calculated from an exact f, the eigenvalues λ_i are exact eigenvalues of the original matrix H. The eigenvectors x_{Ai} give exact components of the corresponding eigenvectors of H, but only n_A of them, namely, those referring to basis elements in S_A. That is, the x_{Ai} are the exact eigenvectors of H belonging to S_A' projected onto the space S_A. Once the eigenvectors x_{Ai} of G_A or $\hat{H}_A$ have been found, the remaining components of the eigenvectors of H are obtained by simple matrix multiplication

$$x_i = \begin{pmatrix} x_{Ai} \\ x_{Bi} \end{pmatrix}, \qquad x_{Bi} = f x_{Ai}, \tag{2.20a}$$

that is, for the whole set,

$$X_{BA} = f X_{AA}. \tag{2.20b}$$

This result follows from Eq. (2.8). Normalization of the eigenvectors of G_A or $\hat{H}_A$ according to (2.16b), namely, $x_{Ai}^{\dagger} g_A x_{Aj} = \delta_{ij}$, corresponds to normalization of the complete eigenvectors of H to unity, namely, $x_i^{\dagger} x_j = \delta_{ij}$. The metric g_A appears because the projected eigenvectors are no longer orthonormal with respect to unity.

Thus, if Eqs. (2.15) can be solved for the matrix f, n_A eigenvalues and eigenvectors of H can be obtained by diagonalizing a matrix of order n_A only. The algorithms presented here are concerned primarily with the determination of f.

2.1. *Alternative Formulas and Properties of Approximate Solutions*

Two distinct formulas for the matrix $\hat{H}_A$ of Eq. (2.18), and for $\hat{H}_B$, are usefully distinguished, namely,

$$\begin{aligned} \hat{H}_A^{(2)} &= g_A^{-1} G_A, \\ \hat{H}_B^{(2)} &= g_B^{-1} G_B, \end{aligned} \tag{2.21a}$$

and

$$\begin{aligned} \hat{H}_A^{(1)} &= H_{AA} + H_{AB} f, \\ \hat{H}_B^{(1)} &= H_{BB} - H_{BA} f^\dagger. \end{aligned} \tag{2.21b}$$

The latter are obtained if Eq. (2.1) is taken as the starting point instead of (2.2), which leads to $HT\hat{X} = T\hat{X}\Lambda$ and to the identity $HT = T(\hat{X}\Lambda\hat{X}^{-1}) = T\hat{H}$, where $\hat{H}$ is to be block diagonal. The two sets of formulas are equivalent only when f is an exact solution of $D(f) = 0$. For an approximate f, it can be shown that

$$\hat{H}_A^{(2)} = \hat{H}_A^{(1)} + g_A^{-1} f^\dagger D(f). \tag{2.22}$$

For a given approximate solution f, the approximation to the exact operator $\hat{H}_A$ given by $\hat{H}_A^{(2)}$ is more accurate than that given by $\hat{H}_A^{(1)}$. Thus a first-order error in f, $\delta f = f^{\text{approx}} - f^{\text{exact}}$, leads to respective errors

$$\delta\hat{H}_A^{(1)} = H_{AB}\, \delta f, \tag{2.23}$$

$$\delta\hat{H}_A^{(2)} = g_A^{-1}[\hat{H}_A{}^\dagger (f^\dagger\, \delta f) - (f^\dagger\, \delta f)\, \hat{H}_A] + O(\delta f^2), \tag{2.24}$$

where the quantities on the right side of (2.24) are the exact ones. The error in $\hat{H}_A^{(1}$ is first order. In the case $n_A = 1$, the commutator-like expression in (2.24) vanishes, so that the error in $\hat{H}_A^{(2)}$ is of second order. This corresponds to the usual property that a first-order error in a trial eigenvector gives only a second-order error in the eigenvalue calculated from the Rayleigh quotient. For $n_A > 1$, the first-order correction to $\hat{H}_A^{(2)}$ does not vanish, in general, but it can be shown generally [10] that the *eigenvalues* of $\hat{H}_A^{(2)}$ are still correct to second order. Thus if f is known only approximately, the eigenvalues and eigenvectors should be determined from $\hat{H}_A^{(2)}$, rather than $\hat{H}_A^{(1)}$, or, equivalently, from Eq. (2.17). This relative insensitivity of $\hat{H}_A^{(2)}$ to errors in f is exploited in the second group of iterative procedures for determining f outlined in Section 3.

It can also be shown [10], on the basis of a theorem of Hylleraas and Undheim [11], and MacDonald [12], that the eigenvalues calculated from $\hat{H}_A^{(2)}$ or G_A, using an

approximate f, are upper bounds to the lowest n_A exact eigenvalues of H, and lower bounds to the highest n_A eigenvalues. The eigenvalues of $\hat{H}_A^{(1)}$ do not have this property.

In terms of $\hat{H}_A^{(1)}$, condition (2.14) on f can be rewritten

$$D^{(1)}(f) = H_{BA} + H_{BB}f - f\hat{H}_A^{(1)} = 0. \tag{2.25}$$

Condition

$$D^{(2)}(f) = H_{BA} + H_{BB}f - f\hat{H}_A^{(2)} = 0 \tag{2.26}$$

is clearly equivalent to (2.25), in the sense that both have the same solutions. This second equation is obtained directly, instead of (2.25), if $T^{-1}HT$ is required to be in block form instead of $T^{\dagger}HT$. In the present context, but not in the more general one in Section 4, it can be shown that $D^{(2)} = g_B^{-1}D^{(1)}$. The two forms (2.25) and (2.26) lead to the two types of algorithms given in the next section.

The magnitude (in a suitable sense) of $D(f)$ is closely related to the exactness of the eigenprojection onto S_A' defined by f. The condition that the projection

$$P' = X^{(A)}X^{(A)\dagger} = \begin{pmatrix} g_A^{-1} & g_A^{-1}f^{\dagger} \\ fg_A^{-1} & fg_A^{-1}f^{\dagger} \end{pmatrix} \tag{2.27}$$

be an exact eigenprojection, is that $[H, P'] = HP' - P'H = 0$. One scalar measure of the deviation of this commutator from zero is the quantity

$$\sigma^2 = \mathrm{tr}([H, P']^2) = \mathrm{tr}\, P'H(1 - P')H, \tag{2.28}$$

which can be regarded as a generalization to a multidimensional case of the variance $\sigma^2 = \langle H^2 \rangle - \langle H \rangle^2$ for a single eigenvector. It can be verified that

$$\sigma^2 = \mathrm{tr}(D^{(2)}(f)\, g_A^{-1}D^{(1)}(f)^{\dagger}) \tag{2.29a}$$

$$= \| g_B^{-1/2}D^{(1)}(f)\, g_A^{-1/2} \|^2, \tag{2.29b}$$

where $\| A \|$ denotes the Hilbert–Schmidt norm, $(\mathrm{tr}\, A^{\dagger}A)^{1/2}$. Since g_A and g_B are positive definite, σ^2 vanishes if and only if $D(f)$ vanishes. It can be shown [10] that σ^2 is the sum of the variances, $\sigma_i^2 = \langle i \mid H^2 \mid i \rangle - \langle i \mid H \mid i \rangle^2$, for the n_A individual eigenvectors calculated from G_A or $\hat{H}_A^{(2)}$. Thus $\sigma^2 = \sum_i \sigma_i^2$ provides an upper bound for the variance of any single eigenvector.

It can also be demonstrated that the gradient of the trace of H over the subspace defined by the projection of Eq. (2.27) is given by

$$\begin{aligned} \partial E/\partial f_{\sigma r}^{*} &= \partial\ \mathrm{tr}\ P'H/\partial f_{\sigma r}^{*} \\ &= (g_B^{-1}D^{(1)}(f)\, g_A^{-1})_{\sigma r}, \end{aligned} \tag{2.30}$$

and is stationary with respect to all variations of f if and only if $D(f) = 0$. Since the trace of H is stationary only when taken over an eigenspace, Eq. (2.30) merely verifies that the vanishing of $D(f)$ is equivalent to an exact uncoupling of S_A. From the invariance of the trace, it must equal the sum of the Rayleigh quotients of the individual eigenvectors of $\hat{H}_A^{(2)}$. Equation (2.30) could be used as the basis for calculaitng a given number of the highest or lowest eigenvalues and eigenvectors of H using a gradient minimization technique. This possibility is not explored here, however.

3. Determination of f

The main computational step is the determination of the matrix f. Among the simplest iterative techniques to apply are those in which Eq. (2.15) is written as a fixed point problem

$$f = \mathscr{F}(f) = \mathscr{A}^{-1}[D(f) + \mathscr{A}f], \tag{3.1}$$

where $\mathscr{A}$ is some nonsingular, possibly f-dependent, superoperator. Successive substitutions $f_{m+1} = \mathscr{F}(f_m)$, starting from an initial guess f_0, give the scheme

$$\begin{aligned} f_{m+1} &= f_m + \delta f_{m+1}, \\ \delta f_{m+1} &= \mathscr{A}^{-1} D(f_m), \qquad m = 0, 1, 2, \ldots, \end{aligned} \tag{3.2}$$

hopefully convergent to a solution of (3.1) and (2.15). If the sequence converges, the convergence will be linear, whatever the choice of $\mathscr{A}$ [8]. Iterative schemes with better than linear convergence properties are impractical because they involve manipulation of unacceptably large amounts of data during each iteration.

When H is diagonally dominant, with the diagonal elements of H_{AA} closely grouped about a value $\lambda_A{}^0$, the simple choice $\mathscr{A} = \lambda_A{}^0 1_B - H_{BB}^{(d)}$ suggests itself, where $H_{BB}^{(d)}$ is the diagonal part of H_{BB}. This gives an iterative scheme,

$$\delta f_{\sigma r} = (\lambda_A{}^0 - H_{\sigma\sigma})^{-1} D_{\sigma r}(f), \tag{3.3}$$

closely related to degenerate perturbation theory. Here, and throughout, Greek letters refer to basis elements in S_B, and Roman letters refer to basis elements in S_A. More generally, for diagonally dominant matrices, the simple choice $\mathscr{A} = 1_B \otimes H_{AA}^{(d)} - H_{BB}^{(d)} \otimes 1_A$ (direct product notation), leads to the iterative scheme

$$\delta f_{\sigma r} = (H_{rr} - H_{\sigma\sigma})^{-1} D_{\sigma r}(f), \tag{3.4}$$

also closely related to perturbation theory.

For more general matrices, however, such iterative schemes may not converge. A better approach is to base the choice of $\mathscr{A}$ on approximations to the appropriate

generalized Newton–Raphson equations, which, if soluble, would give second-order convergence. The Newton–Raphson correction to a trial f is calculated from

$$J(f)\,\delta f = -D(f), \tag{3.5}$$

where J is the Jacobian matrix, consisting of the first derivatives of the elements of $D(f)$ with respect to the elements of f (a special case of the operator $\mathscr{A}$ except for sign). Different procedures are suggested by the Newton–Raphson equations for the two equivalent conditions, $D^{(1)}(f) = 0$, and $D^{(2)}(f) = 0$.

3.1. *Methods Based on* $D^{(1)}(f)$

If f_0 is an approximation to the solution, and δf the exact correction, so that $f = f_0 + \delta f$ is an exact solution of $D^{(1)}(f) = 0$, then it follows from the definition of $D^{(1)}$ that

$$\hat{H}_B^{(1)\dagger}(f_0)\,\delta f - \delta f \hat{H}_A(f) = -D^{(1)}(f_0). \tag{3.6}$$

This is an exact equation for δf. The Newton–Raphson equations,

$$\hat{H}_B^{(1)\dagger}\,\delta f - \delta f \hat{H}_A^{(1)} = -D^{(1)}, \tag{3.7a}$$

differ from (3.6) only in that the exact matrix $\hat{H}_A(f)$ appearing in (3.6) is replaced by the current $\hat{H}_A^{(1)}(f_0)$. That is, the matrix elements of the Jacobian are

$$J_{\rho t,\sigma r} = \partial D_{\rho t}^{(1)}/\partial f_{\sigma r} = (\hat{H}_B^{(1)\dagger})_{\rho\sigma}\,\delta_{rt} - \delta_{\rho\sigma}(\hat{H}_A^{(1)})_{tr}\,. \tag{3.7b}$$

The simplicity, and sparseness of J suggests that the second-order convergent Newton–Raphson method itself might be attractive. In this connection, it will be seen that if $\hat{H}_A^{(1)}$ in (3.5) is approximated by $\hat{H}_A^{(2)}(f_0)$, the resulting system of equations must approach third-order convergence, and in fact, from Eq. (2.24), is third-order convergent when only a single eigenvector is sought ($n_A = 1$) [13]. However, the Newton–Raphson method requires the solution of the system of $n_A n_B$ simultaneous linear equations (3.5) at each iteration. This involves of the order of $n_A{}^3 n_B{}^3$ operations, which is a factor of $n_A{}^2 n_B$ greater than acceptable, even with the improved rate of convergence.

On the other hand, a modified Gauss–Seidel iteration scheme for the solution of Eqs. (3.7a) leads to a simple and useful procedure. In this, the off-diagonal elements of J are ignored, or, equivalently, $\mathscr{A}$ of Eq. (3.1) is taken as the negative of the diagonal part of J, so that the corrections are calculated according to

$$\delta f_{\sigma r} = [(\hat{H}_A^{(1)})_{rr} - (\hat{H}_B^{(1)})_{\sigma\sigma}]^{-1}\, D_{\sigma r}^{(1)}. \tag{3.8}$$

The most efficient procedure, in view of the simplicity of the operators involved, is to change a single element of f at a time, calculating $D_{\sigma r}^{(1)}$ at that time, with continual

updating of $\hat{H}_A^{(1)}$ and $\hat{H}_B^{(1)}$. Where the diagonal elements of H_{AA} are fairly well separated from those of H_{BB}, the normal starting approximation is $f = 0$, in which case the starting approximations to $\hat{H}_A^{(1)}$ and $\hat{H}_B^{(1)}$ are simply H_{AA} and H_{BB}, though any starting f can be tried. After changing a single $f_{\sigma r}$, $\hat{H}_A^{(1)}$ and $\hat{H}_B^{(1)}$ are easily updated because they are linear in f:

$$(\delta \hat{H}_A^{(1)})_{rs} = H_{r\sigma}\, \delta f_{\sigma s} \qquad (r = 1,..., n_A), \tag{3.9}$$

$$(\delta \hat{H}_B^{(1)\dagger})_{\sigma\rho} = -\delta f_{\sigma s}\, H_{s\rho} \qquad (\rho = n_A + 1,..., n). \tag{3.10}$$

Only the diagonal elements $\delta \hat{H}_{\sigma\sigma}^{(1)} = -\delta f_{\sigma s} H_{s\sigma}$ of $\hat{H}_B^{(1)\dagger}$ are in fact required. It is more efficient in this scheme to calculate the elements of $D^{(1)}$ from f and $\hat{H}_A^{(1)}$ as needed, rather than using an updating procedure, which requires storing $D^{(1)}$. The iterative scheme based on Eq. (3.8) will be referred to as "simple diagonal Newton–Raphson" (SDNR). A precise statement of computational details is given in the Appendix.

The idea of the correction $\delta f_{\sigma r}$, calculated in (3.8), is that it should reduce the corresponding $D_{\sigma r}^{(1)}$ approximately to zero. This may be far from the case early in a calculation, if $\delta f_{\sigma r}$ is large. The exact change $\delta f_{\sigma r}$ required to reduce $D_{\sigma r}^{(1)}$ exactly to zero follows from Eq. (3.6), which gives an equation quadratic in $\delta f_{\sigma r}$, namely,

$$[(\hat{H}_A^{(1)})_{rr} - (\hat{H}_B^{(1)})_{\sigma\sigma}]\, \delta f_{\sigma r} + H_{r\sigma}\, \delta f_{\sigma r}^2 = D_{\sigma r}^{(1)}. \tag{3.11}$$

This equation could be useful instead of (3.8) in difficult cases. Iteration of (3.8) for $\delta f_{\sigma r}$, while updating $(\hat{H}_A^{(1)})_{rr}$, but keeping $(\hat{H}_B^{(1)})_{\sigma\sigma}$ and $D_{\sigma r}^{(1)}$ fixed, is equivalent, if convergent, to solving (3.11).

3.2. *Methods Based on $D^{(2)}(f)$*

The operator $\hat{H}_A^{(1)}(f)$ appearing in $D^{(1)}(f)$ must be considered to have errors of the same order as those in f itself. However, for a first-order error in f, the eigenvalues of $\hat{H}_A^{(2)}(f)$, appearing in $D^{(2)}(f)$, are correct to second order, and it is reasonable to treat $\hat{H}_A^{(2)}$ itself as of higher accuracy than f, and to neglect the change in $\hat{H}_A^{(2)}$ in the Newton–Raphson equations appropriate to Eq. (2.26). Because of the inverse matrix g_A^{-1} appearing in $\hat{H}_A^{(2)}$, the exact Jacobian is no longer simple. However, on neglecting $\delta \hat{H}_A^{(2)}$ in comparison to δf, the approximation

$$\partial D_{\sigma r}^{(2)} / \partial f_{\rho t} = J_{\sigma r,\rho t}^{(2)} \simeq H_{\sigma\rho}\delta_{rt} - \delta_{\sigma\rho}(\hat{H}_A^{(2)})_{tr} \tag{3.12}$$

is obtained, which gives the simple equations

$$H_{BB}\, \delta f - \delta f \hat{H}_A^{(2)} = -D^{(2)}(f), \tag{3.13}$$

for δf. Note, in contrast to Eq. (3.6), that this equation involves the original H_{BB} only, and not some modified $n_B \times n_B$ matrix. On the other hand, $\hat{H}_A^{(2)}$ is more com-

plicated to update than $\hat{H}_A^{(1)}$ in (3.7). The change in $\hat{H}_A^{(2)}$ is given exactly, for any change δf, by

$$\begin{aligned}\delta\hat{H}_A^{(2)} &= g_A^{-1(\text{new})} G_A^{(\text{new})} - g_A^{-1(\text{old})} G_A^{(\text{old})} \\ &= g_A^{-1(\text{new})} [\delta G_A - \delta g_A \hat{H}_A^{(2)}] \\ &= g_A^{-1(\text{new})} [\delta f^\dagger (D^{(2)} + H_{BB}\,\delta f - \delta f \hat{H}_A^{(2)}) + W^\dagger \delta f - f^\dagger\,\delta f \hat{H}_A^{(2)}],\end{aligned} \tag{3.14}$$

where

$$W = H_{BA} + H_{BB} f, \tag{3.15}$$

and where all quantities in the last line of (3.14) not explicitly indicated as being new estimates, are the values before updating. An $n_A \times n_A$ matrix inversion is now required for each updating of $\hat{H}_A^{(2)}$ and, as a result, the procedure is efficient only if groups of elements of f are changed simultaneously before updating $\hat{H}_A^{(2)}$. In particular, it is most efficient in application to large matrices, to change entire n_A-dimensional rows of f at one time. For $n_B \gg n_A$, this leads to an algorithm requiring comparable work, per iteration, to algorithm SDNR, that is, or the order of $n_A n_B^2$ operations per iteration. In both approaches, only single columns of the block H_{BB} are required at one time.

Two iterative methods based on Eqs. (3.13) appear useful. The first is the simplest diagonal approximation, which corresponds to taking $\mathscr{A}$ of (3.1) again as the negative of the diagonal part of $J^{(2)}$. This leads to the iteration formula

$$\delta f_{\sigma r} = [(\hat{H}_A^{(2)})_{rr} - H_{\sigma\sigma}]^{-1} D_{\sigma r}^{(2)} \qquad (r = 1,\ldots, n_A). \tag{3.16}$$

When δf is given by (3.16), expression (3.14) for $H_A^{(2)}$ can be simplified somewhat to

$$\delta\hat{H}_A^{(2)} = g_A^{-1(\text{new})} [f_{A\sigma}^{(\text{new})\dagger}\,\delta f_{\sigma A}\,\hat{H}_A^{(2)} + W_{A\sigma}^\dagger\,\delta f_{\sigma A} + (\delta f^\dagger)_{A\sigma}\,\delta f_{\sigma A}\,\hat{H}_A^{(2)d}], \tag{3.17}$$

where $\hat{H}_A^{(2)d}$ is the diagonal part of $\hat{H}_A^{(2)}$, and where $f_{\sigma A}$ and $\delta f_{\sigma A}$ refer to the σth row of f and δf respectively.

The second method is to treat the n_A equations in (3.13) for each fixed σ as a matrix equation. This corresponds to taking $\mathscr{A}$ to be block diagonal, each block being the negative of the diagonal block of $J^{(2)}$ of Eq. (3.12) referring to a row of δf. It yields the iteration formula

$$\delta f_{\sigma A} = D_{\sigma A}^{(2)} [H_{\sigma\sigma} 1_A - \hat{H}_A^{(2)}]^{-1}, \tag{3.18}$$

which, in practice, involves the solution of a system of n_A simultaneous linear equations in n_A unknowns. In this case, the first term in Eq. (3.14) vanishes, so that the updating formula reduces to

$$\delta\hat{H}_A^{(2)} = g_A^{-1(\text{new})} [W^\dagger\,\delta f - f^\dagger\,\delta f \hat{H}_A^{(2)}]. \tag{3.19}$$

This method involves somewhat more computation per sweep through f than the preceding one, but may be expected to converge in fewer overall iterations in certain cases where the off-diagonal elements of H_{AA} are large.

In the case $n_A = 1$, both these methods based on $D^{(2)}(f)$ reduce to the algorithm of Nesbet [4]. We refer to them here as "diagonal generalized Nesbet" (DGN), and "full generalized Nesbet" (FGN), respectively. As discussed in the following section, they are easily adapted to the case of a nonorthogonal basis, with only (roughly) a doubling of computations per sweep.

3.3. *Minimization of the Trace*

To conclude this section, we observe that the Newton–Raphson equations appropriate to the minimization condition $\delta \operatorname{tr}(P'(f)H) = 0$, Eq. (2.30), can be written

$$(g_B^{-1}G_B g_B^{-1})\,\delta f\, g_A^{-1} - g_B^{-1}\,\delta f\,(g_A^{-1}G_A g_A^{-1}) - \bar{D}\,\delta f^\dagger f g_A^{-1} - f g_A^{-1}\,\delta f^\dagger\,\bar{D} = -\bar{D}, \quad (3.20)$$

where $\bar{D}_{\sigma r} = \partial(\operatorname{tr} P'H)/\partial f_{\sigma r}^* = (g_B^{-1}D^{(1)}(f)\,g_A^{-1})_{\sigma r} = (D^{(2)}\,g_A^{-1})_{\sigma r}$. On multiplying left and right by g_B and g_A, respectively, this becomes

$$\hat{H}_B^{(2)\dagger}\,\delta f - \delta f \hat{H}_A^{(2)} = -D^{(1)} + [D^{(1)}g_A^{-1}\,\delta f^\dagger f + f\,\delta f^\dagger\,D^{(2)}]. \quad (3.21)$$

The last term is of higher order than the others, as the minimum is approached, and the remaining equation is of the same type as (3.7a). However, the exact equations (3.20) and (3.21) could be significant for gradient minimization techniques. Note, however, in the exact equation, that the evaluation of $\hat{H}_B^{(2)}$ here involves the inversion of an $n_B \times n_B$ matrix g_B, as well as formation of the product $G_B g_B^{-1}$.

4. Nonorthonormal Basis

A nonorthonormal basis gives rise to the generalized eigenvalue problem

$$Hx_i = \lambda_i S x_i\,, \quad (4.1)$$

where S is the positive definite Gram matrix of the basis elements. Equations (2.2) are replaced by

$$X^\dagger H X = \Lambda, \quad (4.2a)$$

$$X^\dagger S X = 1_n\,. \quad (4.2b)$$

A partitioning procedure, generalizing that of Section 2, leads to formally the same reduced eigenvalue problems as before (Eqs. (2.17)–(2.19)). However, the transformation matrix T is now essentially more complicated,

$$T = \begin{bmatrix} 1_A & h \\ f & 1_B \end{bmatrix}, \quad (4.3)$$

where, as a consequence of Eq. (4.2),

$$h = -(S_{AA} + f^{\dagger}S_{BA})^{-1}(S_{AB} + f^{\dagger}S_{BB}), \tag{4.4}$$

and also the definitions of the quantities appearing in the reduced eigenvalue equations are changed. The metrics, defined by $g = T^{\dagger}ST$, are given by

$$g_A = S_{AA} + S_{AB}f + f^{\dagger}S_{BA} + f^{\dagger}S_{BB}f, \tag{4.5a}$$

$$g_B = S_{BB} + S_{BA}h + h^{\dagger}S_{AB} + h^{\dagger}S_{AA}h, \tag{4.5b}$$

while, from $G = T^{\dagger}HT$,

$$G_A = H_{AA} + H_{AB}f + f^{\dagger}H_{BA} + f^{\dagger}H_{BB}f, \tag{4.6a}$$

$$G_B = H_{BB} + H_{BA}h + h^{\dagger}H_{AB} + h^{\dagger}H_{AA}h, \tag{4.6b}$$

G_B now being changed. $\hat{H}_A^{(2)}$ is still defined as $g_A^{-1}G_A$, but now

$$\hat{H}_A^{(1)} = (S_{AA} + S_{AB}f)^{-1}\,(H_{AA} + H_{AB}f), \tag{4.7a}$$

$$\hat{H}_B^{(1)} = (S_{BB} + S_{BA}h)^{-1}\,(H_{BB} + H_{BA}h), \tag{4.7b}$$

$\hat{H}_A^{(1)}$ and $\hat{H}_A^{(2)}$ still being related by Eq. (2.22). The two equivalent conditions on f are now

$$D^{(1)}(f) = H_{BA} + H_{BB}f - (S_{BA} + S_{BB}f)\,\hat{H}_A^{(1)} = 0, \tag{4.8a}$$

$$D^{(2)}(f) = H_{BA} + H_{BB}f - (S_{BA} + S_{BB}f)\,\hat{H}_A^{(2)} = 0, \tag{4.8b}$$

with $D^{(2)} = (S_{BB} + S_{BA}h)\,g_B^{-1}D^{(1)}$. The gradient of the sum of the n_A Rayleigh quotients with respect to f is given by $D^{(2)}g_A^{-1}$.

The matrix inversions now required to calculate $\hat{H}_A^{(1)}$ and $\hat{H}_B^{(1)}$ mean that the condition $D^{(1)}(f) = 0$ is considerably more complicated than before, and a straightforward generalization of the simple diagonal Newton–Raphson scheme of Section 3 is not useful. Since every element of $\hat{H}_A^{(1)}$ is changed whenever a single element of f is changed, it is no longer efficient to scan through δf, element by element, with constant updating. It is preferrable to change n_A elements at one time, giving a scheme resembling the methods based on $D^{(2)}$. The methods based on $D^{(2)}(f)$ generalize straightforwardly. The approximate Jacobian becomes

$$J^{(2)}_{\sigma r,\rho t} = \partial D^{(2)}_{\sigma r}/\partial f_{\rho t} \simeq H_{\sigma\rho}\delta_{rt} - S_{\sigma\rho}(\hat{H}_A^{(2)})_{tr}\,, \tag{4.9}$$

where, as before, changes in $\hat{H}_A^{(2)}$ are considered to be of higher order and are neglected. This Jacobian is no longer sparse, and if the off-diagonal elements of S_{BB} become large, the convergence of algorithms in which they are ignored, including those given here, may be adversely affected.

The formula for updating $\hat{H}_A^{(2)}$ is now

$$\delta\hat{H}_A^{(2)} = g_A^{-1(\text{new})}[\delta f^\dagger(D^{(2)} + H_{BB}\,\delta f - S_{BB}\,\delta f\hat{H}_A^{(2)}) + W^\dagger\,\delta f - Y^\dagger\,\delta f\hat{H}_A^{(2)}], \tag{4.10}$$

where $W = H_{BA} + H_{BB}f$, as before, and

$$Y = S_{BA} + S_{BB}f. \tag{4.11}$$

Because both W and Y must be calculated, a sweep through δf requires, for $n_B \gg n_A$, approximately twice the computation required in the case of an orthonormal basis ($S = 1_n$).

There are again two plausible ways to calculate δf. As in the method DGN, the matrix $\mathscr{A}$ of Eq. (3.1) can be chosen to be just the negative of the diagonal of the Jacobian, which yields the iteration formula

$$\delta f_{\sigma r} = [S_{\sigma\sigma}(\hat{H}_A^{(2)})_{rr} - H_{\sigma\sigma}]^{-1}\,D_{\sigma r}^{(2)}. \tag{4.12}$$

This choice allows the cancellation of parts of the first and third terms in (4.10), to give

$$\delta\hat{H}_A^{(2)} = g_A^{-1(\text{new})}[W_{A\sigma}^\dagger\,\delta f_{\sigma A} + Y_{A\sigma}^{\dagger(\text{new})}\,\delta f_{\sigma A}\,\hat{H}_A^{(2)} + S_{\sigma\sigma}\,\delta f_{\sigma A}^\dagger\,\delta f_{\sigma A}\,\hat{H}_A^{(2)d}]. \tag{4.13}$$

Alternatively, $\mathscr{A}$ can be chosen block diagonal, with each diagonal block referring to a row of δf, and equal to the negative of the corresponding block of the Jacobian. This yields the iteration formula

$$\delta f_{\sigma A} = D_{\sigma A}^{(2)}[S_{\sigma\sigma}\hat{H}_A^{(2)} - H_{\sigma\sigma}1_A]^{-1}, \tag{4.14}$$

which again, in practice, involves the solution of a system of n_A simultaneous linear equations in n_A unknowns. The first term of $\hat{H}_A^{(2)}$ then vanishes completely, leaving

$$\delta\hat{H}_A^{(2)} = g_A^{-1(\text{new})}[W_{A\sigma}^\dagger\,\delta f_{\sigma A} - Y_{A\sigma}^\dagger\,\delta f_{\sigma A}\,\hat{H}_A^{(2)}]. \tag{4.15}$$

The algorithms based on (4.12)–(4.13) and (4.14)–(4.15) are, respectively, designated nonorthogonal diagonal and full generalized Nesbet (DGNS and FGNS). A precise statement of these two algorithms is given in the Appendix.

5. Computational Results

The algorithms described in Section 3 have been applied to matrices of the type considered by Nesbet [4], in which the off-diagonal elements are unity, and the diagonal elements are the first n odd integers, 1, 3, 5,... . Orders up to 250 were considered, there being no reason, for testing purposes, to consider larger matrices. Permutations of the diagonal elements are equivalent to changes in the basis space S_A, and usually lead to convergence to different eigenspaces S_A'. The calculations were carried out on an IBM 370/168 computer using double-precision arithmetic. The criterion of convergence was based on the Hilbert–Schmidt norm, $\| D\| = (\mathrm{tr} D^\dagger D)^{1/2}$, of the particular form of $D(f)$ used in each method. In practice, a criterion based on the maximum change $\delta f_{\sigma\tau}$ in the elements of f, during an iterative sweep, might also be useful.

For the basic Nesbet matrix, with S_A as the basis space of the lowest (or highest) diagonal elements, all methods converge to give the n_A lowest (or highest)eigenvalues. The number of iterative sweeps varies little with the number of eigenvectors sought, and may either increase or decrease with n_A. In particular, for the highest eigenvalues, or equivalently for the lowest eigenvalues when the off-diagonal elements have changed sign, fewer iterations are required for $n_A > 1$ than for $n_A = 1$, except for DGN. Increasing the order of the matrix generally increases the number of iterations required for $n_A = 1$. When S_A is not the space corresponding to the smallest (or largest) diagonal elements, convergence usually still occurs, but not always.

For convergent calculations it was found, except for the first few iterations in a few cases, that $\log \| D \|$ is very well approximated as a linear function of the iteration number. That is, once the iterative calculation stabilized, convergence was linear in all cases, the value of $\| D \|$ decreasing on the average for each iteration by some constant factor. Table I gives these convergence rates for a number of examples, to illustrate the effects of varying the size of the matrix, the ordering of the diagonal elements, and the differences between diagonal elements of H_{AA} and H_{BB}. In all examples here the basis space S_A is defined by the first n_A basis functions in order, so that reordering of the diagonal elements is equivalent to varying S_A. In addition to the convergence rates for the three algorithms of Section 3, those for the simple perturbation scheme of Eq. (3.4) are also included, and designated SP. Nesbet's algorithm was used to obtain a single eigenvalue of each of these matrices and the convergence rate, as measured by σ, is tabulated for comparison.

The first group of calculations shows the effect of change of the size n, and of the number of eigenvectors sought, when it is the lowest eigenvectors which are sought. The second is for the highest eigenvectors. The third group contains various critical reorderings of diagonal elements. Convergence rates are here, on the whole, less favorable. Note that the different methods may here converge to different subsets of eigenvalues. The fourth group illustrates the suitability of these methods for cases in which the first few diagonal elements are nearly the same, and yet well separated from the rest. In all but one case, the simple perturbation calculation is inferior to the simple diagonal Newton–Raphson algorithm.

TABLE I

Linear[a] Convergence Rates Of the Algorithms In Selected Calculations[b]

n_A	n $(n_A + n_B)$	Diagonal matrix elements in order	Method[c] SDNR	SP	DGN	FGN	Nesbet $n_A = 1$
1	10	1, 3, 5,..., 17, 19[d]	0.24	0.27	0.23	0.23	0.23
1	20	1, 3, 5,..., 37, 39	0.31	0.33	0.30	0.30	0.30
1	250	1, 3, 5,..., 497, 499	0.51	0.52	0.50	0.50	0.50
5	10	1, 3, 5,..., 17, 19	0.22	0.26	0.44	0.19(4)	0.23
5	20	1, 3, 5,..., 37, 39	0.29	0.38	0.37	0.29(4)	0.30
5	250	1, 3, 5,..., 497, 499	0.51	0.55	0.49	0.50(4)	0.50
5	10	19, 17, 15,..., 3, 1	0.23[i]	0.24[i]	0.74(3)[i]	0.24(2)[i]	0.54
5	10	1, 3, 5,..., 11, 9,..., 17, 19	0.33(5)	[l]	0.40[e]	0.30(2)[e]	0.23
5	10	1, 3,..., 11, 9, 7, 13,..., 17, 19	0.55[f]	0.61[f]	0.42[f]	0.31(4)[f]	0.23
5	10	1, 3,..., 11, 13, 7, 9,..., 17, 19	0.31[g]	0.58[g]	0.42[g]	0.999(4)[h]	0.23
5	10	1, 3,..., 13, 11, 9, 7,..., 17, 19	0.31[g]	0.57[g]	0.43[g]	0.999(4)[h]	0.23
5	10	19, 17,..., 9, 11,..., 3, 1	[k]	0.59[j]	0.64[j]	0.23[j]	0.54
5	20	1, 1.1, 1.2, 1.3, 1.4, 11, 13,..., 37, 39	0.18	0.20	0.21	0.16	[n]
5	20	1, 1.1, 1.2, 1.3, 1.4, 1.5, 13, 15,..., 37, 39	0.20	[l]	0.98	0.90	[n]
5	20	1, 1.1, 1.2, 1.3, 1.4, 3, 5,..., 29, 31	0.41	0.56	0.57	[m]	[n]
5	20	1, 1.1, 1.2, 1.3, 1.4, 1.5, 3, 5,..., 27, 29	0.45	[l]	[l]	[l]	[n]

[a] Least-squares calculation of slope of $\log \| D \|$ as a function of iteration number.

[b] The tabulated numbers represent the average factor by which the norm $\| D \|$ is decreased per iteration, once a linear convergence rate is established. S_A is spanned by the first n_A basis functions. All off-diagonal matrix elements are unity.

[c] The number of iterations before linear convergence is established is indicated in brackets to the right of the convergence factor, if not zero.

[d] The eigenvalues of this matrix are 0.386, 2.461, 4.519, 6.573, 8.629, 10.691, 12.766, 14.868, 17.037, 22.072.

[e] Converges to the eigenvalues 0.386, 2.461, 4.519, 6.573, 10.691.

[f] Converges to the eigenvalues 0.386, 2.461, 4.519, 8.629, 10.691.

[g] Converges to the eigenvalues 0.386, 2.461, 4.519, 10.691, 12.766.

[h] Apparently converges to the eigenvalues 0.386, 2.461, 4.519, 10.691, 14.868.

[i] Converges to the eigenvalues 10.691, 12.766, 14.868, 17.037, 22.072.

[j] Converges to the eigenvalues 8.629, 12.766, 14.868, 17.037, 22.072.

[k] $\| D \|$ is oscillatory.

[l] $\| D \|$ is apparently divergent.

[m] $\| D \|$ becomes constant (=4.34) after 25 iterations.

[n] σ becomes constant or increases very slowly after about 50 iterations.

6. Conclusion

A general partitioning approach for the calculation of a small number of eigenvalues and eigenvectors of a large hermitian matrix has been outlined, and three specific algorithms have been given within the context of this approach. Two of these are readily generalized to calculations employing nonorthogonal basis sets. All three require only small sections of the whole matrix at one time and, for all three, the number of operations per iteration is proportional to $n_A n_B^2$, when the number, n_A, of eigenvectors desired is much smaller than the dimension $n = n_A + n_B$ of the matrix. While it is in no way limited to such cases, the present approach is especially useful when the first n_A diagonal elements of the matrix are very similar, though well separated from the remaining diagonal elements.

The computational tests indicate that there is little to choose between the rates of convergence of the three methods. In cases where convergence is not straightforward one method may be more stable, or more rapidly convergent than the others, but there is no clear cut superiority of one in all cases. The simple diagonal Newton–Raphson procedure, based on $D^{(1)}(f)$, is somewhat easier to program efficiently for $n_A > 1$, than the methods based on $D^{(2)}(f)$, and from this standpoint it is particularly attractive. In fact, in most cases, the rates of convergence for this method compare very favorably with those for the other, more complex methods. While this method yields only the approximation $\hat{H}^{(1)}$ directly, a calculation of $\hat{H}^{(2)}$ at the end of the iterative sequence requires only of the order of the time of one iteration. For $n_A = 1$, in the case of an orthonormal basis, SDNR offers an alternative to Nesbet's method, of comparable efficiency.

Appendix

This appendix gives detailed descriptions of the implementation of the algorithms discussed above. In various instances below, especially in the updating cycles, the order in which the computations are done is important. Greek indices refer to basis elements in S_B, Roman indices to basis elements in S_A.

1. *Simple Diagonal Newton–Raphson (SDNR)*

Initialization:

$$f = 0,$$

$$\hat{H}_A^{(1)} = H_{AA},$$

$$\hat{H}_B^{(1)\mathrm{diag}} = H_{BB}^{\mathrm{diag}};$$

then

$$D^{(1)}_{\sigma r} = H_{\sigma r} + \sum_{\rho=1}^{n_B} H_{\sigma\rho} f_{\rho r} - \sum_{t=1}^{n_A} f_{\sigma t} (\hat{H}^{(1)}_A)_{tr},$$

$$\delta f_{\sigma r} = D^{(1)}_{\sigma r} / [(\hat{H}^{(1)}_A)_{rr} - (\hat{H}^{(1)}_B)_{\sigma\sigma}].$$

$r = 1,..., n_A,$
$\sigma = 1,..., n_B.$

Update:

$$(\hat{H}^{(1)}_A)_{sr} \to (\hat{H}^{(1)}_A)_{sr} + H_{s\sigma}\, \delta f_{\sigma r} \qquad (s = 1,..., n_A),$$

$$(\hat{H}^{(1)}_B)_{\sigma\sigma} \to (H^{(1)}_B)_{\sigma\sigma} - \delta f_{\sigma r}\, H_{r\sigma},$$

$$f_{\sigma r} \to f_{\sigma r} + \delta f_{\sigma r}.$$

2. *Diagonal Generalized Nesbet (DGN)*

Initialization:

$$f = 0,$$

$$\hat{H}^{(2)}_A = H_{AA},$$

$$g_A = 1_A;$$

then

$$W_{\sigma r} = H_{\sigma r} + \sum_{\rho=1}^{n_B} H_{\sigma\rho} f_{\rho r},$$

$$D^{(2)}_{\sigma r} = W_{\sigma r} - \sum_{t=1}^{n_A} f_{\sigma t} (\hat{H}^{(2)}_A)_{tr},$$

$$\delta f_{\sigma r} = D^{(2)}_{\sigma r} / [(\hat{H}^{(2)}_A)_{rr} - H_{\sigma\sigma}],$$

$r = 1,..., n_A.$

$\sigma = 1,..., n_B.$

Update:

$$(\delta g_A)_{ts} = \delta f^{\dagger}_{t\sigma} f_{\sigma s} + f^{\dagger}_{t\sigma}\, \delta f_{\sigma s} + \delta f^{\dagger}_{t\sigma}\, \delta f_{\sigma s} \qquad (s, t = 1,..., n_A),$$

$$g_A \to g_A + \delta g_A,$$

$$f_{\sigma r} \to f_{\sigma r} + \delta f_{\sigma r} \qquad (r = 1,..., n_A),$$

$$A_{rs} = f^{\dagger}_{r\sigma} \sum_{t=1}^{n_A} \delta f_{\sigma t}\, (\hat{H}^{(2)}_A)_{ts} + \delta f^{\dagger}_{r\sigma}\, \delta f_{\sigma s}\, (\hat{H}^{(2)}_A)_{ss} + W^{\dagger}_{r\sigma}\, \delta f_{\sigma s}$$

$$(r, s = 1,..., n_A),$$

$$\hat{H}^{(2)}_A \to \hat{H}^{(2)}_A + g_A^{-1} A,$$

3. *Full Generalized Nesbet (FGN)*

Initialization:

$$f = 0,$$

$$\hat{H}^{(2)}_A = H_{AA},$$

$$g_A = 1_A;$$

then

$$W_{\sigma r} = H_{\sigma r} + \sum_{\rho=1}^{n_B} H_{\sigma\rho} f_{\rho r}\,,$$

$$D^{(2)}_{\sigma r} = W_{\sigma r} - \sum_{t=1}^{n_A} f_{\sigma t} (\hat{H}^{(2)}_A)_{tr}\,, \qquad r = 1,\ldots, n_A\,,$$

solve

$$\delta f_{\sigma A}[\hat{H}^{(2)}_A - H_{\sigma\sigma} 1_A] = D^{(2)}_{\sigma A}\,. \qquad \sigma = 1,\ldots, n_B\,.$$

Update:

$$(\delta g_A)_{ts} = \delta f^{\dagger}_{t\sigma} f_{\sigma s} + f^{\dagger}_{t\sigma}\,\delta f_{\sigma s} + \delta f^{\dagger}_{t\sigma}\,\delta f_{\sigma s} \qquad (s, t = 1,\ldots, n_A),$$

$$g_A \to g_A + \delta g_A\,,$$

$$A_{rs} = W^{\dagger}_{r\sigma}\,\delta f_{\sigma s} - f^{\dagger}_{r\sigma} \sum_{t=1}^{n_A} \delta f_{\sigma t}\,(\hat{H}^{(2)}_A)_{ts} \qquad (r, s = 1,\ldots, n_A),$$

$$\hat{H}^{(2)}_A \to \hat{H}^{(2)}_A + g_A^{-1} A,$$

$$f_{\sigma r} \to f_{\sigma r} + \delta f_{\sigma r} \qquad (r = 1,\ldots, n_A),$$

4. *Nonorthogonal Diagonal Generalized Nesbet (DGNS)*

Initialization:

$$f = 0,$$
$$\hat{H}^{(2)}_A = S^{-1}_{AA} H_{AA}\,,$$
$$g_A = S_{AA}\,;$$

then

$$Y_{\sigma r} = S_{\sigma r} + \sum_{\rho=1}^{n_B} S_{\sigma\rho} f_{\rho r} \qquad (r = 1,\ldots, n_A),$$

$$W_{\sigma r} = H_{\sigma r} + \sum_{\rho=1}^{n_B} H_{\sigma\rho} f_{\rho r}\,,$$

$$D^{(2)}_{\sigma r} = W_{\sigma r} - \sum_{t=1}^{n_A} Y_{\sigma t} (\hat{H}^{(2)}_A)_{tr}\,, \qquad r = 1,\ldots, n_A\,.$$

$$\delta f_{\sigma r} = D^{(2)}_{\sigma r} / [(\hat{H}^{(2)}_A)_{rr} S_{\sigma\sigma} - H_{\sigma\sigma}], \qquad \sigma = 1,\ldots, n_B\,.$$

Update:

$$(\delta g_A)_{ts} = \delta f^{\dagger}_{t\sigma}\, Y_{\sigma s} + \delta Y^{\dagger}_{t\sigma} f_{\sigma s} + \delta f^{\dagger}_{t\sigma}\, S_{\sigma\sigma}\, \delta f_{\sigma s} \qquad (t, s = 1,\ldots, n_A),$$

$$g_A \to g_A + \delta g_A\,,$$

$$Y_{\sigma r} \to Y_{\sigma r} + S_{\sigma\sigma}\,\delta f_{\sigma r} \qquad (r = 1,\ldots, n_A),$$

$$A_{rs} = W^{\dagger}_{r\sigma}\,\delta f_{\sigma s} + Y^{\dagger}_{r\sigma} \sum_{t=1}^{n_A} \delta f_{\sigma t}\,(\hat{H}^{(2)}_A)_{ts} + \delta f_{r\sigma}\, S_{\sigma\sigma}\, \delta f_{\sigma s}\,(\hat{H}^{(2)}_A)_{ss}$$
$$(r, s = 1,\ldots, n_A),$$

$$\hat{H}^{(2)}_A = \hat{H}^{(2)}_A + g_A^{-1} A,$$

$$f_{\sigma r} \to f_{\sigma r} + \delta f_{\sigma r} \qquad (r = 1,\ldots, n_A),$$

5. *Nonorthogonal Full Generalized Nesbet (FGNS)*

Initialization:

$$f = 0,$$

$$\hat{H}_A^{(2)} = S_{AA}^{-1} H_{AA},$$

$$g_A = S_{AA};$$

then

$$Y_{\sigma r} = S_{\sigma r} + \sum_{\rho=1}^{n_B} S_{\sigma\rho} f_{\rho r} \qquad (r = 1,\ldots, n_A),$$

$$\left.\begin{aligned} W_{\sigma r} &= H_{\sigma r} + \sum_{\rho=1}^{n_B} H_{\sigma\rho} f_{\rho r}, \\ D_{\sigma r}^{(2)} &= W_{\sigma r} - \sum_{t=1}^{n_A} Y_{\sigma t} (\hat{H}_A^{(2)})_{tr}, \end{aligned}\right\} \quad r = 1,\ldots, n_A;$$

solve

$$\delta f_{\sigma A} [S_{\sigma\sigma} \hat{H}_A^{(2)} - H_{\sigma\sigma} 1_A] = D_{\sigma A}^{(2)}. \qquad \sigma = 1,\ldots, n_B.$$

Update:

$$(\delta g_A)_{ts} = \delta f_{t\sigma}^{\dagger} Y_{\sigma s} + \delta Y_{t\sigma}^{\dagger} f_{\sigma s} + \delta f_{t\sigma}^{\dagger} S_{\sigma\sigma} \delta f_{\sigma s} \qquad (s, t = 1,\ldots, n_A),$$

$$g_A \to g_A + \delta g_A,$$

$$A_{ts} = W_{t\sigma}^{\dagger} \delta f_{\sigma s} - Y_{t\sigma}^{\dagger} \sum_{r=1}^{n_A} \delta f_{\sigma r} (\hat{H}_A^{(2)})_{rs} \qquad (t, s = 1,\ldots, n_A),$$

$$\hat{H}_A^{(2)} \to \hat{H}_A^{(2)} + g_A^{-1} A,$$

$$f_{\sigma r} \to f_{\sigma r} + \delta f_{\sigma r} \qquad (r = 1,\ldots, n_A).$$

ACKNOWLEDGMENT

Thanks are due to David Calder for testing certain algorithms considered at an earlier stage of this work. The support of the National Research Council of Canada is gratefully acknowledged.

REFERENCES

1. E. R. DAVIDSON, *J. Comp. Phys.* **17** (1975), 87.
2. I. SHAVITT, C. F. BENDER, A. PIPANO, AND R. P. HOSTENY, *J. Comp. Phys.* **11** (1973), 90.
3. I. SHAVITT, *J. Comp. Phys.* **6** (1970), 124.
4. R. K. NESBET, *J. Chem. Phys.* **43** (1965), 311.
5. M. G. FELER, *J. Comp. Phys.* **14** (1974), 341.
6. W. BUTSCHER AND W. E. KAMMER, *J. Comp. Phys.* **20** (1976), 313.

7. B. Roos, The Configuration Interaction Method, *in* "Computational Techniques in Quantum Chemistry and Molecular Physics" (G. H. F. Diercksen, B. T. Sutcliffe, and A. Veillard, Eds.), Reidel, Boston, 1975.
8. L. B. Rall, "Computational Solution of Nonlinear Operator Equations," Wiley, New York, 1969.
9. J. A. R. Coope, *Mol. Phys.* **18** (1970), 571.
10. J. A. R. Coope and D. W. Sabo, to appear.
11. E. A. Hylleraas and B. Undheim, *Z. Physik* **65** (1930), 759.
12. J. K. L. MacDonald, *Phys. Rev.* **43** (1933), 830.
13. J. F. Traub, "Iterative Methods for the Solution of Equations," p. 220, Prentice–Hall, Englewood Cliffs, N. J., 1964.