

Improved Algorithms for the Lowest Few Eigenvalues and Associated Eigenvectors of Large Matrices

CHRISTOPHER W. MURRAY, STEPHEN C. RACINE, AND ERNEST R. DAVIDSON

Department of Chemistry, Indiana University, Bloomington, Indiana 47405

Received February 15, 1991; revised November 6, 1991

Some modifications of Davidson's eigenvalue algorithm are discussed and their performances on a number of test cases are assessed. They are found to offer improvements over the original algorithm. A method for solving the equations stemming from the *quasi degenerate variational perturbation theory* is presented. Solutions can be obtained as simply as with the eigenvalue algorithm for both ground and excited states. © 1992 Academic Press, Inc.

1. INTRODUCTION

The configuration interaction (CI) method is used to calculate electronic wavefunctions for the ground state and excited states of atoms and molecules. It leads to the formation of a large, real, symmetric matrix for which the lowest eigenvalues and associated eigenvectors must be found. Generally about 1-5% of the matrix elements are non-zero and less than 10 eigenvectors are required. The dimension of the matrix is usually larger than 10^4 so the non-zero matrix elements are stored on disk. For large scale calculations, with matrix dimensions greater than 10^5 , the matrix may be too large to store on disk and the elements must be recalculated as they are needed [1]. Matrices of dimension greater than 10^6 have been used and the present record size is 10^9 [2]. In these cases, a simple iterative method is preferred for calculating the eigenvectors since the non-zero matrix elements are accessed in an essentially random order. The Davidson algorithm [3] is the most widely used approach for matrices of this type, and it will be described in the next section. A number of improvements and amendments to the usual implementation of the algorithm will be discussed and some results will be presented.

Quasi degenerate variational perturbation theory (QDVARPT) [4] is an alternative method for calculating electronic wavefunctions. The third section contains a description of how the equations resulting from this method may be solved for the ground state and excited states. The similarities and differences between this approach and the

Davidson algorithm will be discussed. The section also includes results that compare the performance of the algorithms for similar matrices.

2. THE EIGENVALUE PROBLEM

The eigenvalue equation is

$$\mathbf{H}\mathbf{c}_k = \lambda_k \mathbf{c}_k \quad (1)$$

for which only the K lowest eigenvalues and eigenvectors are required. K is very much less than the dimension of the matrix, N . For the sake of clarity it will now be assumed that only the lowest eigenvalue is desired, and the problem of obtaining the higher ones will be considered separately.

Given some guess to the eigenvector, \mathbf{x} , perturbation theory can be used to give an improved eigenvector, $(\mathbf{x} + \delta)$, where δ satisfies

$$(\mathbf{H} - \lambda\mathbf{I})\delta = -(\mathbf{H} - \lambda\mathbf{I})\mathbf{x}. \quad (2)$$

Suppose the eigenvalue, λ , is approximated by the Rayleigh quotient defined by

$$\rho = \mathbf{x}^T \mathbf{H} \mathbf{x} / \mathbf{x}^T \mathbf{x}. \quad (3)$$

For Eq. (2) to be useful, some simple approximation to the inverse of $(\mathbf{H} - \rho\mathbf{I})$ must be used. The Davidson method approximates the matrix by its diagonal, \mathbf{D} , and Eq. (2) can be rewritten to give

$$\delta = -(\mathbf{D} - \rho\mathbf{I})^{-1} (\mathbf{H} - \rho\mathbf{I})\mathbf{x}. \quad (4)$$

Other estimates of the inverse of $(\mathbf{H} - \rho\mathbf{I})$ have been suggested by Morgan and Scott [5]. At this point, $(\mathbf{x} + \delta)$ could be used as the new \mathbf{x} in Eq. (4) and this procedure could be repeated until convergence had been reached. Unfortunately, this approach has poor convergence characteristics.

The method can be improved by coupling it with one of the desirable features of the Lanczos method [6]. The trial vector, \mathbf{x}^n , at some iteration n , is expanded in a linear combination of orthogonal vectors.

$$\mathbf{x}^n = \sum_{i=0, n} \alpha_i \mathbf{b}^i. \quad (5)$$

The subspace spanned by the \mathbf{b} vectors will be referred to as the \mathbf{b} space, and the matrix composed of the \mathbf{b} vectors as \mathbf{B} . The coefficients α_i are found by minimising the Rayleigh quotient of \mathbf{x}^n , leading to a projected eigenvalue equation

$$\mathbf{B}^T \mathbf{H} \mathbf{B} \alpha = \rho \alpha. \quad (6)$$

This yields an upper bound to the exact eigenvalue. The correction to this optimal vector is obtained via Eq. (4) and the correction vector is orthogonalised to the \mathbf{b} space. The \mathbf{b} space is then supplemented by this vector and the whole procedure is repeated until convergence is reached. At each iteration, the algorithm requires the formation of the product $\mathbf{H} \mathbf{b}^n$, and for large matrices, this step takes most of the time. The step also requires the storage in main memory of the full length vectors \mathbf{b}^n and $\mathbf{H} \mathbf{b}^n$, and additionally, the application of Eq. (4) and (5) necessitate the storage on disk of all previous \mathbf{b} vectors in the subspace and their products with \mathbf{H} .

The iterative process can be considered to have converged when the weight of the latest addition to the \mathbf{b} space in Eq. (5) drops below some threshold. Alternatively, convergence can be assessed by the size of the norm of the residual vector, \mathbf{r} ,

$$\mathbf{r} = (\mathbf{H} - \rho \mathbf{I}) \mathbf{x} \quad (7)$$

which yields a lower bound to the exact eigenvalue. In applications with CI matrices, a typical threshold for both tests is 10^{-4} .

The extension to the calculation of higher roots is straightforward. The higher roots produced by the subspace diagonalisation are necessarily approximations to the higher roots of the full matrix. The k th root of the full matrix can be obtained by successive improvement of the vector derived from the k th eigenvector of the subspace matrix. This approach is guaranteed to be successful when the estimates of the $k - 1$ lower roots produced by subspace matrix diagonalisation are smaller than the k th exact eigenvalue. This means that reasonable approximations to the lower eigenvectors must be contained in the \mathbf{b} space.

When all the eigenvalues up to a certain number are required, there is some flexibility in the order in which one optimises them. Liu has pointed out [7] that since one of the major bottlenecks in the solution of (1) is reading in from disk or recalculating the matrix elements, it is advan-

tageous to add more than one vector at a time to the \mathbf{B} matrix. This approach will be referred to as the block Davidson method because of its similarity to the block Lanczos method. For each \mathbf{b} vector added, it is necessary to store two vectors in main memory, and so in general, not all δ_k , $k = 1, \dots, K$, can be considered on every iteration. We add as many vectors as memory constraints allow, and cycle through the unconverged roots in turn until all the desired eigenvectors have been converged. Our experience is that the procedure offers considerable saving in CPU time over the "one root at a time" approach, and this will be illustrated by examples later in this section.

The choice of initial guess will affect the convergence rate of the algorithm. There is a danger that the method may sometimes converge to the wrong eigenvector if poor guesses are provided. The extreme of this situation would be where the matrix contained hidden symmetry and the starting vector has zero overlap with the desired root; in this case, the algorithm will fail. Theoretically, except in these symmetry controlled failures, the method will always converge to the right state, provided that the vector is converged tightly enough; but in practice, sufficiently tight convergence thresholds are not always used. It is therefore important that initial guesses be chosen that have reasonably large overlaps with the lowest eigenvectors of the matrix.

A common method of generating initial guesses would be by diagonalisation of a small submatrix of \mathbf{H} . Another approach would be to choose unit vectors which were thought to have contributions in the desired eigenvectors. The simplest way of choosing the unit vectors would be based on the size of the diagonal elements of \mathbf{H} . Alternatively, second-order perturbation theory can be used to improve on this estimate of the eigenvalues, but this requires a pass through the whole matrix so it is more expensive. One advantage of using unit vectors is that one is less likely to run into problems associated with hidden symmetry. However, none of these choices guarantees convergence to the lowest solutions and a more practical approach is to simply ask the algorithm to find more roots than are needed. We routinely investigate more roots than are wanted and converge them to a soft convergence threshold (0.005). The unwanted higher roots are then rejected and the remaining roots fully converged. It is noted that if one requires an eigenvalue of unknown index but known zeroth-order eigenvector, one can convergence to it selectively by always choosing that eigenvectors of the subspace matrix that gives the largest overlap with the initial guess [8].

The algorithm requires the storage of every \mathbf{b} vector and its product with the full matrix, and for large matrices they must be stored on disk and accessed at every iteration. The list of vectors must be truncated periodically to restrict the storage used by the vectors and to reduce the IO associated with their manipulation. The list is usually contracted to the current approximations to the roots required, and this is

equivalent to restarting the procedure with improved initial guesses. The truncation is expected to inhibit convergence because each vector in the \mathbf{b} space will contribute to the best trial vector at each iteration. If the \mathbf{b} space is truncated, the trial vector at the next iteration will not benefit from the minimisation of the Rayleigh quotient in the full \mathbf{b} space. Van Lenthe and Pulay have pointed out recently [9] that, when searching for the lowest vector, truncation down to two vectors may be performed at every iteration without significant reduction in the speed of convergence. This startling conclusion comes from the theory of the conjugate gradient method [10, 11].

The conjugate gradient method and the related method of conjugate directions can be used to solve linear equations of the type

$$\mathbf{A}\mathbf{c} = \mathbf{k}, \quad (8)$$

where \mathbf{k} is some constant vector and \mathbf{A} is a real, symmetric, positive definite matrix [10]. It is an iterative process in which, \mathbf{x}^i , the approximation to the solution vector at some iteration, i , is expanded in a subspace, \mathbf{P} :

$$\mathbf{x}^i = \mathbf{x}^0 + \alpha_0 \mathbf{p}^1 + \dots + \alpha_{i-1} \mathbf{p}^{i-1}. \quad (9)$$

In the expansion, \mathbf{x}^0 is the initial guess at the solution. The vectors \mathbf{p} (together with \mathbf{x}^0) are chosen to be mutually conjugate in the sense that for different vectors \mathbf{p}^j and \mathbf{p}^k with j and k less than i ,

$$(\mathbf{p}^j)^T \mathbf{A} \mathbf{p}^k = 0. \quad (10)$$

The coefficients in Eq. (10) are found by minimising the error function, $f(\mathbf{x}^i)$, where

$$f(\mathbf{x}) = (\mathbf{c} - \mathbf{x})^T \mathbf{A} (\mathbf{c} - \mathbf{x}). \quad (11)$$

If the process is judged not to have converged, a new vector, \mathbf{p}^i , is found from the residual, \mathbf{r}^i , by making \mathbf{r}^i conjugate to the vectors in \mathbf{P} . The residual is defined by

$$\mathbf{r}^i = \mathbf{k} - \mathbf{A} \mathbf{x}^i. \quad (12)$$

The basic results of the conjugate gradient theory are that this choice of residual is orthogonal to the subspace \mathbf{P} , and that condition (11) can be satisfied for the new \mathbf{p} vector without explicitly considering all previous \mathbf{p}^j . It follows that minimisation of $f(\mathbf{x}_i)$ in the subspace \mathbf{P} is equivalent to minimisation in the restricted space of \mathbf{x}^{i-1} and \mathbf{p}^{i-1} .

This is the rationale for truncating the vector list to two vectors rather than one. It was argued [7] that in the latter stages of the Davidson procedure the eigenvalue is essentially constant, and so the eigenvalue equation is like a linear

equation system. For the ground state, the appropriate linear equation is obtained from Eq. (2) as

$$\mathbf{Q}^T (\mathbf{H} - \rho \mathbf{I}) \mathbf{Q} \mathbf{d} = \mathbf{Q}^T (\mathbf{H} - \rho \mathbf{I}) \mathbf{x}, \quad (13)$$

where \mathbf{Q} is the $N \times (N-1)$ projection into the subspace orthogonal to \mathbf{x} , $\mathbf{Q} \mathbf{d} = \delta$, and intermediate normalisation has been used so $\delta^T \mathbf{x} = 0$. The matrix $\mathbf{Q}^T (\mathbf{H} - \rho \mathbf{I}) \mathbf{Q}$ will always be definite provided that

$$\rho < (\lambda_1 + \lambda_2)/2. \quad (14)$$

If the eigenvalue is sufficiently converged that it can be assumed to be constant, this condition should be satisfied. Following work by Wormer *et al.* [10], Van Lenthe and Pulay showed that in this case the Davidson algorithm is formally equivalent to the conjugate gradient method. Consequently, truncation to two vectors will not impede convergence. When forming the new \mathbf{B} matrix, we set \mathbf{b}^1 equal to the current guess to the eigenvector, \mathbf{x}^i , and set \mathbf{b}^2 equal to that part of the previous iteration's guess to the vector, \mathbf{x}^{i-1} , that is orthogonal to \mathbf{x}^i . (These vectors span the same space as the vectors \mathbf{x}^{i-1} and \mathbf{p}^{i-1} referred to in the preceding discussion.) The underlying assumption of constant eigenvalue is justified because the minimisation of ρ in the \mathbf{b} subspace is variational, and the eigenvalue converges much faster than the vector.

It was proposed [7] that truncation to two vectors would be useful in very large matrix diagonalisations where the matrix is never stored but is recalculated. In such cases, it is conceivable that there may only be enough disk space to store a few vectors and products, in which case the method offers a considerable advantage. We have taken a slightly different view because truncation will still inhibit convergence in so far as the eigenvalue is not constant during the iterative process. Therefore in the general case, one should use as much disk space as one has available, provided that the IO associated with the manipulation of the vectors does not start to become competitive with the time taken accessing or calculating the matrix. In line with the conjugate gradient method, the \mathbf{B} matrix should be truncated to two vectors per root needed, rather than just one. In a fixed amount of disk space, this means that truncation will be performed more often, and in cases where the eigenvalue is not approximately constant, it could actually slow down convergence. However, our experience has been that this form of truncation is more effective than the original approach.

The question arises as to whether this truncation can be applied successfully to higher roots. To examine this consider the linear equation analogous to (13) for higher roots. \mathbf{Q} will now be an $N \times (N-k)$ projection into the subspace orthogonal to the k current guesses to the k lowest

eigenvectors. The condition that the matrix $\mathbf{Q}^T(\mathbf{H} - \rho_k \mathbf{I})\mathbf{Q}$ be positive definite is now

$$\rho < (\lambda_k + \lambda_{k+1})/2 - \sum_{i=0, k-1} (\rho_i - \lambda_i)/2, \quad (15)$$

where ρ_i is the Rayleigh quotient estimate of the i th eigenvalue, λ_i . The condition is much stricter for higher roots than for the first root and in general requires that lower roots must be well converged. Truncation to two vectors would therefore be desirable when roots are obtained one at a time with the lower ones computed first, but as has been discussed previously, block methods are a more efficient way of obtaining the first K roots of a large matrix. We truncate to two vectors per unconverged root in our block Davidson algorithm and hope that the lower roots converge more quickly and are sufficiently converged so that the crucial matrix for the k th root is positive definite. This means that at some iteration, i , where truncation is performed, the new \mathbf{b} space spans the set $\{\mathbf{x}_k^i, \mathbf{x}_k^{i-1}\}$ for all $k \leq K$ corresponding to converged roots, together with the set $\{\mathbf{x}_k^i\}$ for all $k \leq K$ corresponding to converged roots. If the lower roots are not sufficiently converged, the only harm done is that the extra vector included in the \mathbf{b} set will take up space which could perhaps have been used by a more effective vector.

One of the key steps in the Davidson algorithm is the minimisation of the Rayleigh quotient in the orthogonal subspace generated in the iterative procedure. Another expansion is possible based on the minimisation of the least square error in the residual vector [12–13]. The subspace eigenvalue equation then takes the form

$$\mathbf{M}\alpha = \sigma\alpha, \quad (16)$$

where the matrix \mathbf{M} is given by

$$M_{ij} = [(\mathbf{H} - \rho\mathbf{I})\mathbf{b}^i]^T (\mathbf{H} - \rho\mathbf{I})\mathbf{b}^j \quad (17)$$

and the new, extrapolated guess, \mathbf{x}^e , is defined through the lowest eigenvector of \mathbf{M} by

$$\mathbf{x}^e = \mathbf{B}\alpha. \quad (18)$$

The extrapolation procedure is specific to one eigenvector (through ρ), so in a block procedure the diagonalisation must be performed more than once. However, since it is our implicit assumption that the subspace matrix is very much smaller than the original one, this not a drawback. The choice to make for ρ in Eq. (17) is not clear cut. To be truly consistent, ρ should be evaluated as a function of \mathbf{x}^e , but this would make the solution of Eq. (16) more troublesome. We obtain ρ from the variational minimisation of the Rayleigh quotient in the current \mathbf{b} space as embodied by Eq. (6).

One of the attractive features of the least squares method is that it concentrates on convergence of the vector rather than the eigenvalue. The latter is quickly converged in the variational minimisation of ρ , so the method could be used to complement the Davidson algorithm rather than as an alternative. As a check on the reliability of the least squares extrapolation, the Rayleigh quotient of the extrapolated vector is calculated. If the energy increases by too much the extrapolation is rejected. This policy came from experience with the method; it was observed that the method was more prone to converge to higher states than the Davidson method. This was especially noticeable when more than one root was sought. Also we found that the method converged more slowly when used instead of the block Davidson method, but in contrast, superior convergence was observed if the least squares extrapolation was performed periodically.

The exact combination that we employed was the following. The block Davidson algorithm was used until the root in question had converged beyond a certain threshold. For CI matrices the spacing of the eigenvalues is typically 0.1 to 0.5 and a good threshold here appears to be 5×10^{-3} . The extrapolation was performed one out of every three times for roots that satisfied the first criteria. If ρ changed by more than 10^{-3} as a result of the least square step, the extrapolation was rejected.

Table I illustrates the performance of these various improvements to the algorithm for a small CI matrix where we were interested in obtaining 10 roots. The block Davidson method is seen to be superior to the “one root at a time” approach, despite the fact that considerably more products of the vector on the matrix are performed. Truncation to two vectors per root rather than one resulted in a 20% reduction in the number of iterations, and the use of

TABLE I

Comparison of Different Modifications of the Davidson Method

Method	Number of iterations	Number of matrix vector products	CPU time (seconds on IBM 3090 120S)
A	392	392	3740
B	101	537	2960
C	83	470	2580
D	85	389	2270
E	62	367	2040

Note. The matrix is from a restricted CI of F_2^+ with a (12s, 8p, 6d, 2f, g) basis set. The matrix dimension is 3803 and 10 eigenvectors were found. A. Original Davidson algorithm with each root obtained separately. B. Block Davidson method. C. Method B with truncation to two vectors following conjugate gradient approach. (Truncation occurred when 100 vectors were stored on disk.) D. Method B with least squares extrapolation. (See text for details of how often the extrapolation was performed.) E. Method C with least squares extrapolation.

occasional least squares extrapolation caused a further 20% increase in efficiency. This data shows that all the changes outlined here should be employed with the Davidson algorithm to yield maximum efficiency. The reduction from 392 iterations for the original method to 62 iterations for the block Davidson method with least squares extrapolation and truncation to two vectors is very encouraging. For large matrices where the computational time is controlled by how many times elements are retrieved from disk or are recalculated, these improvements would result in considerable savings.

Although the Davidson algorithm was developed in the context of CI calculations, it is a general purpose algorithm when only a few eigenvectors are wanted for larger matrices. Its performance is generally superior to the Lanczos method. To illustrate this and to test further some of the modifications to the Davidson method that we have out-

lined previously, we have generated some arbitrary and reproducible matrices and used the different methods to obtain eigenvalues. The first family of matrices were of dimension 10^6 with all off-diagonal elements being 0 or -1 . The non-zero matrix elements occurred for i or $j \leq 10$, for $|i-j| \leq 2$, and for $|i-j| \geq 10^6 - 12$. The non-zero matrix elements were regenerated on every iteration rather than stored on disk. The diagonal elements, H_{nn} , $n = 1, \dots, 10^6$, were chosen to be (a) \sqrt{n} , (b) n , or (c) n^2 . The matrix can be solved analytically for the lowest root if the diagonal elements are constant and this allowed us to easily check that the matrix was programmed correctly. For these choices of diagonal elements, the second-order perturbation theory estimate of the lowest eigenvalue gives a series that (a) diverges like \sqrt{N} , (b) diverges like $\ln(N)$, and (c) converges, where N is the dimension of the matrix. As shown in Table II, the algorithm converges in all three cases, but the difficulty of convergence reflects the behaviour of the perturbation series. The Lanczos algorithm, on the other hand, was unable to produce even one eigenvalue after 50 iterations in two of the cases. The block diagonal method is again seen to reduce the number of iterations considerably, although this saving in time is not pronounced since in this example the production of matrix elements is very quick. Similarly, the truncation to two vectors results in a large reduction in the number of iterations for case (a), the only case where truncation is performed. The change of initial guess and the use of occasional least squares extrapolation do not result in significant savings.

TABLE II
Different Diagonalisation Methods Attempted on the
First Test Matrix Which Is Described in the Text

Diagonal elements	Method	Number of roots	Number of iterations	Number of products	Time
Sqrt(n)	A	1	5	5	240
Sqrt(n)	B	1	6	6	290
Sqrt(n)	A	5	20	72	4840
Sqrt(n)	C	5	20	70	4710
Sqrt(n)	D	5	52	52	6590
Sqrt(n)	E	5	31	81	5022
Sqrt(n)	F	5	20	70	5640
n	A	5	5	21	860
n	B	1	> 50		
n	C	5	6	24	1170
n	D	5	14	14	1180
n	F	5	5	21	980
n^2	A	5	4	17	
n^2	B	1	> 50		
n^2	C	5	4	18	
n^2	D	5	11	11	
n^2	F	5	4	17	
<i>First five eigenvalues</i>					
Sqrt(n)	-2859.34966	0.192313	0.881275	1.401652	1.830419
n	-95.485788	2.297765	3.367710	4.419722	5.464841
n^2	-0.251141	4.023443	9.014725	16.011851	25.010810

Note. The matrix has a dimension of 10^6 . The times are in seconds on an IBM RISC series/6000 model 540 computer. A. Guess obtained from diagonalising 10 by 10 matrix; truncation to two vectors per root when 25 vectors and products are on disk; block Davidson method. B. Lanczos method. C. As method 1 but initial guess obtained from the five lowest diagonal elements of matrix. D. As method 1 but only one vector per iteration is operated on with the matrix. E. As method 1 but truncation to one vector per root is performed. F. As method 1 but least squares extrapolation is performed periodically.

TABLE III

Different Diagonalisation Methods Attempted on
Direct Product Matrices Which Are Described in the Text

Dimension	Method	Number of roots	Number of iterations	Number of products	Time (seconds)
1048576	A	1	12	12	842000
1048576	B	1	> 10^a		
65536	A	3	26	58	18000
65536	C	3	52	52	33000
65536	D	3	29	64	20000
<i>First few eigenvalues</i>					
1048576	194306.64				
65536	13517.539	17379.774	17591.649		

Note. A. Guess obtained from lowest diagonal elements; truncation to two vectors per root when 19 vectors and products are on disk; block Davidson method. B. Lanczos method. C. As method A but only one vector per iteration is operated on with the matrix. D. As method A but truncation to one vector per root is performed.

^a The Lanczos method was converging very slowly indeed, and with each iteration costing 19 hours, it was decided to terminate the job after 10 iterations.

A second family of matrices was formed from the direct product of symmetric four by four matrices:

$$\mathbf{A}_L = \begin{bmatrix} 3 + 0.1L & 0.1 & 0.2 & 0.3 \\ 0.1 & 4 + 0.1L & 0.0 & 0.0 \\ 0.2 & 0.0 & 5 + 0.1L & 0.0 \\ 0.3 & 0.0 & 0.0 & 6 + 0.1L \end{bmatrix}. \quad (19)$$

Matrices of dimension 4^J were formed from \mathbf{A}_L , $L = 0, \dots, J-1$ with $(10/16)^J$ of the elements being filled with nonzero values. These matrices are much less sparse and are more difficult to form than those in the previous example. Products of the eigenvalues of the appropriate \mathbf{A}_L matrices are the eigenvalues of the direct product matrices. The $J = 8$ and $J = 10$ matrices were studied and the results are shown in Table III. Again the present algorithm worked without difficulty while the Lanczos algorithm converged very slowly and was aborted. The truncation to two vectors gives a modest improvement over the original truncation approach.

3. SOLUTION TO QDVARPT EQUATIONS

The CI method does not scale properly with the number of electrons and, for large systems, alternative methods are preferred. One of the simplest of the alternative methods is QDVARPT [4], which leads to a set of non-linear equations that need to be solved. Other methods are similar to QDVARPT and efficient methods for obtaining the lowest solution to the equations derived from these approaches have been given by other authors [14, 15]. Here our aim is, first, to describe how we have chosen to solve the equations for the lowest state, pointing out the similarities to the Davidson algorithm and, second, to show how our algorithm can be easily extended to the calculation of the first few roots of the equations.

In the QDVARPT method, the equation to be solved is of the form

$$\begin{bmatrix} \mathbf{H}_{pp} - \lambda \mathbf{I} & \mathbf{H}_{pq} \\ \mathbf{H}_{qp} & \mathbf{H}_{qq} - \lambda_0 \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{C}_p \\ \mathbf{C}_q \end{bmatrix} = \mathbf{0}. \quad (20)$$

λ is the root to be found, and λ_0 is given by $(\mathbf{C}_p)^T \mathbf{H}_{pp} \mathbf{C}_p$. The problem is solved with intermediate normalisation so $(\mathbf{C}_p)^T \mathbf{C}_p$ is 1. The matrix is partitioned into blocks and the \mathbf{C}_p part of the vector is chosen so as to contain the dominant part of the solution vector. The space spanned by the \mathbf{C}_p vector is referred to as the reference space, and its dimension is usually very much less than of the full matrix. Equation (20) can be rewritten as

$$\mathbf{H}_{pp} \mathbf{C}_p + \mathbf{H}_{pq} \mathbf{C}_q = \lambda \mathbf{C}_p \quad (21)$$

and

$$(\lambda_0 \mathbf{I} - \mathbf{H}_{qq}) \mathbf{C}_q = \mathbf{H}_{qp} \mathbf{C}_p \quad (22)$$

Equation (22) can be written in the form of an effective eigenvalue equation

$$\{\mathbf{H}_{pp} + \mathbf{H}_{pq}(\lambda_0 \mathbf{I} - \mathbf{H}_{qq})^{-1} \mathbf{H}_{qp}\} \mathbf{C}_p = \lambda \mathbf{C}_p, \quad (23)$$

where the matrix in curly brackets will be called \mathbf{H}^{eff} .

Given some trial \mathbf{C}_p and \mathbf{C}_q , perturbation theory can be used on Eq. (22) to give a correction δ to the \mathbf{C}_q vector. This is analogous to the procedure used to obtain Eq. (2) and (4) and yields

$$\delta = -(\mathbf{D}_q - \lambda_0 \mathbf{I})^{-1} (\mathbf{H}_{qp} \mathbf{C}_p + \mathbf{H}_{qq} \mathbf{C}_q - \lambda_0 \mathbf{C}_q), \quad (24)$$

where \mathbf{D}_q is the diagonal of the matrix \mathbf{H}_{qq} . At every iteration a new δ will be produced by this approach and, as in the Davidson procedure, the whole set of δ 's will provide a suitable expansion space for \mathbf{C}_q . As before, the vectors in the set will be called \mathbf{b} space and the matrix composed of these vectors will be referred to as \mathbf{B} . \mathbf{C}_q can be expanded in an equivalent way to Eq. (5) as

$$\mathbf{C}_q \simeq \mathbf{B}\alpha, \quad (25)$$

where α is a vector of coefficients to be determined. Replacing for \mathbf{C}_q in Eq. (22), premultiplying by \mathbf{B}^T and rearranging gives

$$\alpha = \{\mathbf{B}^T(\lambda_0 \mathbf{I} - \mathbf{H}_{qq})\mathbf{B}\}^{-1} \mathbf{B}^T \mathbf{H}_{qp} \mathbf{C}_p. \quad (26)$$

It is now convenient to define \mathbf{Z} by

$$\mathbf{Z} = \{\mathbf{B}^T(\lambda_0 \mathbf{I} - \mathbf{H}_{qq})\mathbf{B}\}^{-1} \mathbf{B}^T \mathbf{H}_{qp} \quad (27)$$

so that

$$\alpha = \mathbf{Z}\mathbf{C}_p, \quad (28)$$

in which case, the effective Hamiltonian can be written simply as

$$\mathbf{H}^{\text{eff}} = \mathbf{H}_{pp} + \mathbf{H}_{pq} \mathbf{B}\mathbf{Z}. \quad (29)$$

Thus at each iteration, the new \mathbf{Z} matrix is calculated and then the effective eigenvalue problem is solved to give the updated \mathbf{C}_p . Equations (25) and (28) are then used to give \mathbf{C}_q , from which a correction is derived using perturbation theory (Eq. (24)). The correction is included in the \mathbf{b} space and the iterative procedure is continued until convergence is reached. We apply two convergence criteria simultaneously: the change in λ between iterations must be less than 10^{-6} ;

and for the latest addition to the expansion space $\delta^T \delta$ must be less than 10^{-8} .

The similarities between the Davidson algorithm and the one outlined in this section are obvious. The C_q vector is expanded in the space of the initial guess and perturbatively determined corrections to subsequent approximations to the vector. One difference is that we have chosen not to orthogonalise the expansion space, but this is purely a matter of taste.

The algorithm outlined above also allows us to obtain excited state solutions to Eq. (21) for QDVARPT. A "block" method could be used to solve the problem although it would be more difficult to set up than for the eigenvalue problem because λ_0 will be differently defined for each state and Eq. (20) will need to be solved separately for each state. However, for large matrices the saving in the IO or the recalculation of the matrix would justify the block method. For development purposes, we have chosen to solve for each root separately. In this case, it is important when trying to optimise a higher root that lower vectors (or good approximations to them) are contained in the expansion space. This can be seen by comparing the effective eigenvalue equation, (23), with the projected eigenvalue equation that is actually solved in the algorithm and is given by

$$\left[\mathbf{H}_{pp} + \mathbf{H}_{pq} \mathbf{B} \{ \mathbf{B}^T (\lambda_0 \mathbf{I} - \mathbf{H}_{qq}) \mathbf{B} \}^{-1} \mathbf{B}^T \mathbf{H}_{qp} \right] \mathbf{C}_p = \mathbf{E} \mathbf{C}_p. \quad (30)$$

Diagonalisation of the matrix in Eq. (30) will give good energies for the k lowest states only if the \mathbf{b} space contains good approximations to the C_q vectors for those k states. If this is not true then the ordering of the states on energy will not be reliable and it will not be possible to obtain the k th root of the exact problem in a controlled manner. Table IV contains some results illustrating the performance of the algorithm for excited states. The results are encouraging since they show that higher states can be studied with these methods in a similar manner to the CI problem.

As with the Davidson method, it becomes necessary to truncate the expansion space when it becomes too large or unwieldy. If λ , λ_0 , and C_p are assumed to be constant during the iterative process, the theory of conjugate gradients can be used to justify retaining both the current iteration's guess at the desired root and the previous iterations guess at this root. The argument in the last paragraph emphasises the importance of also putting the lower roots in the truncated space. The assumption that λ , λ_0 , and C_p are constant is reasonable because in the later iterations, the algorithm will mainly be improving the vector, C_q . This truncation procedure is acceptable for both ground and excited states, provided that the matrix $(\mathbf{H}_{qq} - \lambda_0 \mathbf{I})$ is positive definite.

TABLE IV

Comparison of the Eigenvalue Problem with QDVARPT

Method	Number of iterations	Number of matrix-vector products	CPU time (seconds on IBM 3090 120S)
A	62	62	5050
B	17	64	3090
C	82	82	7000

Note. The matrix is from a MRCI with 25 reference configurations on F_2^+ using a $(12s, 8p, 6d, 2f, g)$ basis set. The matrix dimension is 52847 and five eigenvectors were found. A. Original Davidson algorithm with each root obtained separately. B. Block Davidson method with least squares extrapolation. (See text for details of how often the extrapolation was performed.) C. QDVARPT calculation with algorithm outlined in text. For methods A–C truncation was performed when 40 vectors had been stored on disk. Method A truncated to one vector per root whilst methods B and C truncated to two vectors per root.

This requires that the eigenvalues of the \mathbf{H}_{qq} block be higher than λ_0 for all the states of interest, and in most of the problems we are concerned with, this condition is obeyed.

4. CONCLUSION

The Davidson algorithm for large matrix diagonalisation can be converged more quickly by periodic use of least squares extrapolation. The truncation procedure suggested by Pulay and Van Lenthe has been shown to be effective when applied to excited state calculations within the framework of the block Davidson method. These modifications represent an improvement over the traditional approach.

The algorithm given here for the solution of the QDVARPT equations is as efficient as matrix diagonalisation. It can be applied routinely to both the ground state and excited states of molecules and the truncation procedure based on the conjugate gradient method is effective in both cases.

ACKNOWLEDGMENT

This work was supported by a grant from the National Science Foundation.

REFERENCES

1. B. O. Roos, *Chem. Phys. Lett.* **15**, 153 (1972).
2. J. Olsen, P. Jørgensen, and J. Simons, *Chem. Phys. Lett.* **169**, 463 (1990).

3. E. R. Davidson, *J. Comput. Phys.* **17**, 87 (1975); E. R. Davidson, *Comput. Phys. Commun.* **53**, 49 (1989); E. R. Davidson, in *Methods in Computational Molecular Physics*, edited by G. H. F. Diercksen and S. Wilson (Reidel, Dordrecht, 1983).
4. R. J. Cave and E. R. Davidson, *J. Chem. Phys.* **89**, 6798 (1988).
5. R. B. Morgan and D. S. Scott, *SIAM J. Sci. Stat. Comput.* **7**, 817 (1986).
6. C. Lanczos, *J. Res. Nat. Bur. Stand.* **45**, 255 (1950).
7. B. Liu, *Numerical Algorithms in Chemistry: Algebraic Methods*, edited by C. Moler and I. Shavitt, LBL-8158 (Lawrence Berkeley Laboratory, Livermore, CA, 1978).
8. W. Butscher and W. J. E. Kammer, *J. Comput. Phys.* **20**, 13 (1976).
9. J. H. van Lenthe and P. Pulay, *J. Comput. Chem.* **11**, 1164 (1990).
10. M. R. Hestenes and E. Stiefel, *J. Res. Nat. Bur. Stand.* **49**, 498 (1952).
11. P. E. S. Wormer, F. Visser, and J. Paldus, *J. Comput. Phys.* **48**, 23 (1982).
12. M. G. Feler, *J. Comput. Phys.* **14**, 341 (1974).
13. D. M. Wood and A. Zunger, *J. Phys. A* **18**, 1343 (1985).
14. R. Ahlrichs and P. Scharf, in *Ab Initio Methods in Quantum Chemistry I*, edited by K. P. Lawley (Wiley, Chichester, 1987).
15. R. Ahlrichs, *Comput. Phys. Commun.* **17**, 31 (1979).