A method for calculating the extreme eigensolution of a real symmetric matrix of high order

1980 J. Phys. A: Math. Gen. 13 2369

(http://iopscience.iop.org/0305-4470/13/7/019)

J. Phys. A: Math. Gen. **13** (1980) 2369–2374. Printed in Great Britain

A method for calculating the extreme eigensolution of a real symmetric matrix of high order

T Z Kalamboukis

Department of Computing Science, The University, Glasgow G12 8QQ, UK

Received 3 July 1979, in final form 14 November 1979

Abstract. A simple method for calculating the extreme eigensolution of a real symmetric matrix of high order, alternative to Davidson's method, is investigated and compared with other similar methods.

1. Introduction

In shell model calculations on atoms large sparse real symmetric matrices are produced and their lowest lying or highest lying eigenvalue and the corresponding eigenvector are required.

Davidson (1975) introduced a method which is based on restricting the matrix A $(n \times n)$ to a p-dimensional subspace U and approximating its eigenvalues and the corresponding eigenvectors by solving the eigenproblem for the matrix $A_p = U_p^* A U_p$ of order $p \times p$, where $U_p = (b_1, \ldots, b_p)$ is an $n \times p$ orthonormal matrix with columns consisting of the vectors derived in the course of the process.

In the present work we propose a method for the calculation of the extreme eigenvalues and their eigenvectors which restricts the basis vectors b_i to a two-dimensional subspace. The method can be considered as the restriction of Davidson's method to a two-dimensional subspace. In some respects the present method reduces to the minimum all the numerical and computational difficulties of the original Davidson method.

One part of Davidson's method which takes a considerable amount of the computational time and extra storage is the orthogonalisation process of the basis vectors b_i, $i = 1, \ldots, p$. These vectors, as well as the vectors Ab_i, are kept in auxiliary store in order to reduce the computational effort in the following steps. The transfer times from and to the main memory, however, occupy a large part of the computer time.

In every iteration we have to orthogonalise the pth basis vector with respect to the $(p-1)$ foregoing ones. But because of the round-off errors introduced, by cancellation in the subtraction step of the Gram–Schmidt process, the method may break down (Kalamboukis 1979) and it may be necessary to reorthogonalise these vectors. Another step of Davidson's method which takes considerable time is the solution of the eigenproblem for the matrix A_p, of order $p \times p$, in every iteration.

The present method is easy to program and overcomes all the disadvantages mentioned. The small amount of computation which it requires per iteration and the fact that no vectors need to be kept in auxiliary store make it much faster than Davidson's algorithm.

0305-4470/80/072369+06$01.50

The method is similar to the coordinate relaxation method (Faddeev and Fadeeva 1963, Schwarz 1974), which also uses a two-vector subspace, apart from the fact that in the present method all the components of the trial vector are changed simultaneously. The convergence rate of the algorithm has been speeded up considerably, in analogy to successive over-relaxation applied to the coordinate relaxation method, by a systematic under-relaxation. In this way we achieved a great improvement of the convergence rate for values of the relaxation parameter ω in (0, 1). Indeed from our numerical examples we see that for a suitable value of ω we have reduced the number of iterations by half. In spite of the lack of a theoretical estimate of the optimal value for ω, we shall report some numerical examples which show the convergence rate of the algorithm as a function of the relaxation parameter.

A similar method based on restricting the Lanczos method (Lanczos 1950) to a two-dimensional subspace but without using the under-relaxation technique is proposed by Berger *et al* (1977).

The present method has been compared (Kalamboukis 1979) with the coordinate relaxation method and is found to be much faster in the case of diagonally dominant matrices with closely spaced eigenvalues.

Deflation (Shavitt *et al* 1973) has been applied with this method to find some of the interior eigenvalues near the end of the spectrum.

In the following sections we describe our algorithm in more detail and present a proof of the convergence. Finally several numerical results are presented to illustrate how the method works in practice.

2. Description of the method

Consider the eigenvalue problem

$$Ax = \lambda x \tag{2.1}$$

where A is a real symmetric matrix of high order. Let λ_i denote the eigenvalues of (2.1) numbered in ascending order:

$$\lambda_1 < \lambda_2 < \ldots < \lambda_n.$$

In the following we shall restrict the problem to finding only the lowest (λ_1) or the highest (λ_n) eigenvalue and the corresponding eigenvector.

Let μ be the index i for which a_{ii}, $i = 1, \ldots, n$, takes its minimum value. To start the algorithm, if no good approximation of the required eigenvector is available, we take as starting vector b_1 the unit vector e_μ with one in the μth position and zero elsewhere. Then we find a new vector $b_2 = Ab_1 - \lambda b_1$ ($\lambda = b_1^* Ab_1$), and the 2×2 generalised eigenvalue problem

$$\tilde{A}y = \lambda \tilde{B}y \tag{2.2}$$

is solved, where $\tilde{a}_{ij} = b_i^* Ab_j$ and $\tilde{b}_{ij} = b_i^* b_j$, $i, j = 1, 2$. In theory b_2 is orthogonal to b_1 and when b_2 is normalised $\tilde{B}$ is the 2×2 unit matrix. However, in practice b_2 is not exactly orthogonal to b_1; the introduction of $\tilde{B}$ into (2.2) is a way of avoiding the orthogonalisation of b_2 to b_1 which would otherwise be necessary. The lowest eigenvalue of (2.2), that is the smallest root of the quadratic equation

$$(1 - \tilde{b}_{12}^2)^2 - (\tilde{a}_{11} + \tilde{a}_{22} - 2\tilde{a}_{12}\tilde{b}_{12}) + (\tilde{a}_{11}\tilde{a}_{22} - \tilde{a}_{12}^2) = 0, \tag{2.3}$$

is taken as a new approximation of λ and a linear combination of the vectors b_1, b_2, namely

$$y_1 b_1 + y_2 b_2,$$

is taken as a new starting vector and is used to generate a subsequent 2×2 generalised eigenproblem. The process continues until the lowest eigenvalue and the corresponding eigenvector have converged. The scalars y_1, y_2 are the two components of the eigenvector belonging to the lowest eigenvalue of the system (2.2). For the calculation of (y_1, y_2) we have assumed $y_1 = 1$ and thus

$$y_2 = (\tilde{a}_{12} - \lambda \tilde{b}_{12})/(\lambda - \tilde{a}_{22}). \tag{2.4}$$

The process may be summarised in the following simple algorithm†.

Initialisation: Choice of the starting vector b_1
Calculate $b_2 = Ab_1 - \lambda b_1$ and normalise
Iteration: While $\|b_2\| > \epsilon$ or $|y_2| > \epsilon$ do
(a) Form Ab_2
(b) Form the interaction matrices $\tilde{A}$, $\tilde{B}$ and solve the generalised eigenvalue problem $\tilde{A}y = \lambda \tilde{B}y$. Select λ, $y = (y_1, y_2)$
(c) $b_1 := y_1 b_1 + y_2 b_2$, $\|b_1\| = 1$
(d) $Ab_1 := y_1(Ab_1) + y_2(Ab_2)$ (2.5)
(e) Form $b_2 = Ab_1 - \lambda b_1$, $\|b_2\| = 1$.

To accelerate the convergence rate of the algorithm described, we shall incorporate the relaxation factor by multiplying (2.4) by $\omega \in (0, 1)$ in step (b) of the iteration.

From (2.5) we see that the vector $Ab_1^{(k+1)}$, where k denotes the iteration number, can be computed recursively so that each step requires essentially the computation of the matrix vector multiplication $Ab_2^{(k)}$ (step (a)).

For diagonally dominant matrices with off-diagonal elements small compared to the separations of the diagonal elements ($\max_{i,j} |a_{ij}/(a_{ii} - a_{jj})| < 0{\cdot}01$) we can use Davidson's perturbation correction in estimating the vector b_2 to accelerate the convergence, i.e.

$$b_2 \leftarrow b_2/(\lambda - a_{ii}). \tag{2.6}$$

The successive values of $\lambda^{(k)}$ as $k \to \infty$ form a monotonically decreasing sequence and since this is a sequence of Rayleigh quotients it is bounded below by the lowest eigenvalue and therefore is convergent (see Appendix). In practice it does converge to the lowest eigenvalue. It follows that the off-diagonal elements of $\tilde{A}$, $\tilde{B}$ tend to zero. Also $y_2^{(k)} \to 0$ as $k \to \infty$, so $|y_2^{(k)}|$ can be used as a measurement for the accuracy of the eigenvectors of A.

To find some of the interior eigenvalues near the end of the spectrum the deflation method (Shavitt *et al* 1973) has been applied with the present method to a modified matrix which possesses λ_k as the lowest eigenvalue. If $(k-1)$ eigenvalues and the corresponding eigenvectors are known, then

$$A_k = A + \sum_{i=1}^{k-1} \phi_i x_i x_i^*,$$

† An alternative way would be to orthogonalise the vectors b_1, b_2 by the Gram–Schmidt process and replace (2.2) with the solution of the 2×2 single eigenproblem $\tilde{A}x = \lambda x$, where $\tilde{A}$ is as in (2.2).

where x_i are orthonormal eigenvectors of A and ϕ_i are suitably chosen scalars, possesses λ_k as the lowest eigenvalue. The quantities involving the matrix A_k in the algorithm are the matrix vector multiplication and the formation of matrix $\tilde{A}$. These calculations, however, can be formed in an easy manner without forming explicitly the matrix A_k.

If instead of the lowest eigenvalue of the 2×2 generalised eigenproblem (2.2) we take in every iteration the highest one, then we have convergence to the highest eigenvalue of A.

3. Numerical results and conclusions

In this section we shall describe a few examples to illustrate the utility of the present method.

Example 1. In this example three Hamiltonian matrices have been tested, one for ^{20}Ne, of order 640, a slightly different one for ^{20}Ne, also of order 640, with single particle energies equal to zero, and for ^{23}Na, of order 876. These matrices were tested for different values of the relaxation parameter ω. In figure 1 we give a demonstration of the convergence of the method as a function of ω. For a comparison with Davidson's method, the present method for ^{23}Na converged after 104 iterations with $\omega = 0{\cdot}87$ and $|y_2| = 0{\cdot}7\times10^{-8}$. The CPU time was 13 min in an IBM 370/145 computer. Davidson's method converged after 54 iterations with $\|q\| = 0{\cdot}5\times10^{-7}$. The CPU time for Davidson's procedure was 78 min. These CPU times were obtained without attempted programming optimisation.

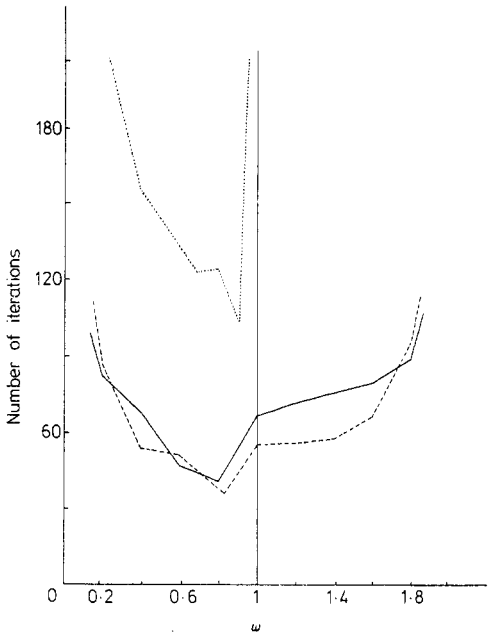

Figure 1. The convergence rate of the present algorithm as a function of the relaxation parameter ω. Continuous line represents ^{20}Ne, broken line ^{20}Ne with single particle energies 0 and dotted line ^{23}Na.

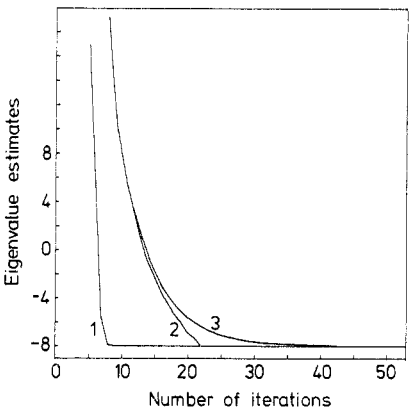

Figure 2. Convergence diagram for example 2. Curve (1) represents Davidson's method, (2) the present method with $\omega = 0{\cdot}9$ and (3) with $\omega = 1{\cdot}0$.

Example 2. A 600×600 random symmetric matrix was treated. In figure 2 we illustrate the convergence diagram for Davidson's method (curve 1), the present method with $\omega = 0{\cdot}9$ (2) and with $\omega = 1$ (3).

Example 3. A 100×100 diagonally dominant matrix with close-together diagonal elements was constructed (Kalamboukis 1979) for a comparison of the present method with the coordinate relaxation method. The matrix was tested several times, varying the dominance of the diagonal elements. From these tests, the coordinate relaxation method is faster for general matrices (table 1, large values of d) while for diagonally dominant matrices ($d < 0{\cdot}01$) the present method is superior using Davidson's perturbation correction (2.6), the purpose of which is to accelerate the convergence (Kalamboukis 1979). (In table 1 the last three entries have been obtained using (2.6).)

For the cases with diagonally dominant matrices the present method compares favourably with Davidson's method (Kalamboukis 1979).

Table 1. Comparison of the present method with coordinate relaxation for different values of $d = \max_{i,i} |a_{ij}/(a_{ii} - a_{jj})|$.

d	Present method ($\omega_{\text{optimum}\,0{\cdot}85}$)		Coordinate relaxation ($\omega_{\text{optimum}\,1{\cdot}2}$)	
	Iterations	$\lvert y_2 \rvert$	Iterations	$\lvert y_2 \rvert$
1·0	28	$0{\cdot}5 \times 10^{-7}$	14	$0{\cdot}3 \times 10^{-8}$
0·1	34	$0{\cdot}7 \times 10^{-7}$	17	$0{\cdot}5 \times 10^{-8}$
0·01	55	$0{\cdot}9 \times 10^{-7}$	15	$0{\cdot}4 \times 10^{-8}$
0·001	11	$0{\cdot}8 \times 10^{-7}$	175	$0{\cdot}5 \times 10^{-6}$
0·0001	5	$0{\cdot}5 \times 10^{-9}$	181	$0{\cdot}3 \times 10^{-3}$
0·00001	3	$0{\cdot}4 \times 10^{-9}$	181	$0{\cdot}9 \times 10^{0}$

From these examples we see that the present method might take more iterations sometimes than Davidson's method, but still be much faster, since the amount of work involved per iteration is much less than that of Davidson's method. This is apparent from example (1) for ^{23}Na, comparing the CPU times. Also we do not have to keep any vectors in auxiliary store and transfer them to the main memory, which saves a lot of computational time, and the sparseness of the matrix is fully taken into account. The matrix is represented in a matrix–vector multiplication form in a subroutine so zeros do not appear in the multiplication.

From all our numerical examples and other examples tested we have noted that for values of $\omega \in (0{\cdot}8, 0{\cdot}9)$ the convergence rate is considerably increased, so a single value of ω such as 0·85 would work well in all the cases. The computational work compares very favourably with the other methods we have discussed. The total number of multiplications per iteration is $(z + 12)n$, where z is the average number of non-zero elements per row of the matrix A. The algorithm can easily be extended to the generalised eigenvalue problem.

In conclusion, the simplicity of programming, the small amount of work per iteration and the good convergence rate for near-degenerate eigenvalues are the main factors in favour of the present method for finding the extreme eigenvalues and their eigenvectors of large sparse real symmetric matrices.

Acknowledgment

I wish to express my thanks to Dr J Haselgrove and Dr R R Whitehead for their helpful discussions and remarks on this work.

Appendix

It is easy to show that the vectors b_i, $i = 1, 2$, are theoretically orthogonal. So the matrix $\tilde{B}$ should be the unit matrix. Suppose that at the kth iteration we have

$$b_1^{(k+1)} = y_1^{(k)} b_1^{(k)} + y_2^{(k)} b_2^{(k)} = U^{(k)} y$$

where $U^{(k)} = (b_1^{(k)}, b_2^{(k)})$. Then

$$\tilde{a}_{11}^{(k+1)} = y^* \tilde{A}^{(k)} y = \lambda_1^{(k)} \tag{A1}$$

and $\lambda_1^{(k)} = \lambda_1 + t_1$ where λ_1 is the exact lowest eigenvalue of A. In the following we shall determine the matrix $\tilde{A}^{(k+1)}$ and hence $\lambda_1^{(k+1)}$. Suppose that

$$b_1^{(k+1)} = cx_1 + \sum_{r \neq 1} e_r x_r \tag{A2}$$

where x_r are the exact eigenvectors, $\|x_r\| = 1$, and $c^2 + \Sigma\ e_r^2 = 1$. From (A1) and (A2) it follows that $t_1 = \Sigma\, e_r^2 (\lambda_r - \lambda_1)$.

To find $\lambda_1^{(k+1)} = \lambda_1 + \gamma$, say, it will be convenient to use the matrix $C = \tilde{A}^{(k+1)} - \lambda_1 I$ and hence to find γ directly. Some straightforward algebra shows that

$$c_{11} = t_1, \qquad c_{12} = s, \qquad c_{22} = (1/s^2)(t_3 - t_1 t_2) - t_1$$

where $s = t_2 - t_1^2$ and $t_k = \Sigma\, e_r^2 (\lambda_r - \lambda_1)^k$. So γ will be the lowest eigenvalue of C,

$$\gamma = t_1 - t_2^2/t_3 + O(e^4).$$

Hence $0 < \gamma < t_1$ and therefore $\lambda_1^{(k+1)} < \lambda_1^{(k)}$, which proves the argument of § 2 that the sequence of $\lambda_1^{(k)}$ as $k \to \infty$ is convergent.

It is clear that λ_1 is the value to which the $\lambda_1^{(k)}$ converges, for convergence implies that $t_2^2/t_3 = 0$, and hence that the error is zero, $t_1 = 0$ and finally $\gamma = 0$. Although we have assumed that λ_1 is the lowest eigenvalue, the argument could apply to any eigenvalue. In practice, however, as with the other methods which seek a stationary value of the Rayleigh quotient, it is the extreme eigenvalue which is reached. In any case, we attempt to start the process with an estimate as close to that value as possible.

References

Berger W, Miller H G, Kreuzer K-G and Dreizler R M 1977 *J. Phys. A: Math. Gen.* **10** 1089–96
Davidson E R 1975 *J. Comp. Phys.* **17**
Faddeev D K and Faddeeva V N 1963 *Computational Methods of Linear Algebra* (San Francisco: W H Freeman)
Kalamboukis T Z 1979 *Ph.D. Thesis* Glasgow University
—— 1980 *J. Phys. A: Math. Gen.* **13** 57–62
Lanczos C 1950 *J. Res. Nat. Bur. Stand.* **45**
Schwarz H R 1974 *Comput. Meth. in Appl. Mech. and Engng* **3** 11–28
Shavitt I, Bender C F, Pipano A and Mosteny R P 1973 *J. Comp. Phys.* **11** 90–108