Introduction to Part II

The Algebraic Eigenvalue Problem

by J. H. WILKINSON

1. Introduction

The standard algebraic eigenvalue problem, the determination of the non-trivial solutions of $Ax = \lambda x$, is one of the most fascinating of the basic problems of numerical analysis. In spite of the simplicity of its formulation many algorithms are required in order to deal efficiently with the wide range of problems which are encountered in practice.

A variety of factors enter into the determination of the most efficient algorithm for the solution of a given problem; the following are perhaps the most important.

i) Eigenvalues *and* eigenvectors may be required or the eigenvalues only.

ii) The *complete* set of eigenvalues and/or eigenvectors may be required or only a comparatively small number. In the latter case one may require, for example, the k smallest, the k largest, the k nearest to a given value or all those lying within a given range.

iii) The matrix may be symmetric or unsymmetric. In the former case we may include more general eigenproblems which can be reduced to the standard problem with a symmetric matrix.

iv) The matrix may be *sparse*, either with the non-zero elements concentrated on a narrow band centred on the diagonal or alternatively they may be distributed in a less systematic manner. We shall refer to a matrix as *dense* if the percentage of zero elements or its distribution is such as to make it uneconomic to take advantage of their presence.

v) For very large matrices economy of storage may well be the paramount consideration in selecting an algorithm.

vi) The results may be required to only a modest accuracy or high precision may be vital.

We remind the reader that in the algorithms presented in this book no reference is made to the use of a backing store such as a magnetic tape or disc. For this reason the storage requirements in practice may be different from what they appear to be. For example if eigenvalues and eigenvectors of a Hessenberg matrix H are found using procedure *hqr* followed by inverse iteration it is necessary to retain a copy of H while the *hqr* reduction is performed. In practice this copy could be written on a backing store without inconvenience, since it is not required until *hqr* is completed.

2. List of Procedures

To facilitate reference to the procedures they are listed below in alphabetical order, together with the chapters in which they are described. Sets of related procedures are grouped together.

balance, balbak, II/11
bandrd, II/8
bisect, II/5
bqr, II/7
comeig, II/17
comhes, combak, II/13
comlr, II/16
comlr 2, II/15
cxinvit II/18
dirhes, dirbak, II/13
dirtrans, II/15
eigen, II/12
elmhes, elmbak, II/13
elmtrans, II/15
hqr, II/14
hqr2, II/15
imtql 1, imtql 2, II/4
invit, II/18
jacobi, II/1
orthes, ortbak, II/13
ortrans, II/15
ratqr, II/6
reduc 1, reduc 2, rebak a, rebak b, II/10
ritzit, II/9
symray, I/6
tql 1, tql 2, II/3
tred 1, tred 2, tred 3, trbak 1, trbak 3, II/2
tristurm, II/18
unsray, I/6

3. Real, Dense, Symmetric Matrices

The computation of the eigensystem of a real symmetric matrix is much simpler than that of a general matrix since the eigenvalues of the former are always well determined.

3.1. Calculation of All the Eigenvalues with or without Eigenvectors

When all the eigenvalues and eigenvectors are required two programs based on the given procedures may be recommended.

a) The Method of Jacobi

The method of Jacobi is the *most elegant* of those developed for solving the complete eigenproblem. The procedure *jacobi* may be used to find all the eigen-

values with or without the eigenvectors. It is an extremely compact procedure and considerable care has been taken to ensure that both eigenvalues and eigenvectors are of the highest precision attainable with the word length that is used. It is not adapted for finding a few selected eigenvalues and eigenvectors and other procedures are generally more efficient for such a purpose.

The computed system of eigenvectors is always orthogonal, almost to working accuracy, and the orthogonality is in no way impaired by the presence of multiple or pathologically close eigenvalues. The accuracy of the orthogonality relation should not be taken to imply a corresponding accuracy in the individual eigenvectors themselves, but there are many applications in which orthogonality is of paramount importance.

b) Householder Tridiagonalization and the QL Algorithm

The *most efficient* program for finding all the eigenvalues alone or all the eigenvalues and eigenvectors is a combination of the Householder tridiagonalization and the QL algorithm. It is usually faster than *jacobi*, in particular if eigenvalues only are required. In this case the optimum combination is *tred 1* and *tql 1*; if economy of storage is paramount then *tred 3* should replace *tred 1* since in the former the original symmetric matrix is stored as a linear array of $\frac{1}{2}n(n+1)$ elements. If both values and vectors are required the optimum combination is *tred 2* and *tql 2*. In each case the appropriate combination is several times faster than *jacobi* except when n is fairly small, say less than ten. *tred 2* and *tql 2* are remarkably effective from the point of view of storage, only $n^2+O(n)$ locations being required for the complete eigenproblem. The eigenvectors are accurately orthogonal independent of the separations.

The procedures *imtql 1* and *imtql 2* may replace *tql 1* and *tql 2* respectively. The speeds of the two sets of procedures are about the same and when the tridiagonal matrix has elements which diminish rapidly in absolute value as one moves down the diagonal the procedures give the smaller eigenvalues to higher accuracy. When *tql 1* and *2* are used on such matrices they should be orientated so that the smaller elements occur first along the diagonal, if necessary by reflecting in the secondary diagonal. These comments imply that the set *imtql 1* and *2* is preferable to the set *tql 1* and *2* and in practice this appears to be true. However there remains the danger that with *imtql 1* and *2* small subdiagonal elements may inhibit convergence, possibly as a result of underflow, but no such examples have been encountered in practice.

When a procedure is needed to solve the complete eigenproblem of a matrix of modest order as part of a larger main program, *jacobi* might well be preferred because of its compactness. (It is used in *ritzit* (see 5.) for this reason.)

3.2. Calculation of Selected Eigenvalues with or without Eigenvectors

When a comparatively small number of eigenvalues is required the methods of the previous section become inefficient. However it should be emphasized that the combinations *tred 1* and *tql 1* or *imtql 1* and *tred 2* and *tql 2* or *imtql 2* are very effective and when more than, say, 40% of the eigensystem is required the relevant combination will usually provide the optimum solution. Unfortunately,

the QL transformation with the standard shift is not well adapted to finding a few specific eigenvalues, and it seems to be difficult to design an aesthetically satisfying shift technique which finds a number of the smallest eigenvalues (a common requirement).

The more effective procedures for finding specified eigenvalues are also based on a preliminary Householder reduction to tridiagonal form using *tred 1* or *tred 3*. Two of the published procedures are then relevant.

a) Givens' Sturm Property Procedure

The procedure *bisect* is based on the Sturm sequence property of the leading principal minors of $T-\lambda I$, where T is the tridiagonal matrix. The published procedure has been designed to find eigenvalues numbered *m1* to *m2*, where they are assumed to be ordered so that $\lambda_1 \leqq \lambda_2 \leqq \cdots \leqq \lambda_n$. Simple modifications to the basic procedure may readily be designed to find, for example, all the eigenvalues in a given interval or the k eigenvalues nearest to a given value. Indeed the virtue of this algorithm is its extreme flexibility. (See also 3.2(c) for a further use of bisection.)

Since the localisation of individual eigenvalues proceeds by repeated bisection the accuracy may be readily varied. In fact the procedure could be speeded up by changing to a method with superlinear convergence after an eigenvalue has been located in an interval. (Modified successive linear interpolation and extrapolation is particularly effective). However, when the matrix has clusters of eigenvalues bisection will, in any case, be pursued until these eigenvalues have been quite closely located. This means that *bisect* is actually faster when there are multiple roots or tight clusters than when they are well separated.

b) Rational QR Transformation with Newton Shift

In the procedure *ratqr*, the normal shift technique of the QR algorithm is modified to a Newton correction such that convergence to the largest or smallest eigenvalues takes place. (With the usual QR shift technique the order in which the eigenvalues are found is largely unpredictable, though there is a tendency to find first the eigenvalue nearest to an eigenvalue of the 2×2 matrix in the bottom right-hand corner.) The shift technique also makes it possible to devise a square-root free version of the QR algorithm. Convergence is not so fast as with *tql 1,2* or *imtql 1,2*, but when a small number of the largest or smallest eigenvalues is required, *ratqr* is considerably more efficient. However, when the required eigenvalues are clustered, *bisect* will be faster since Newton's method has only linear convergence for multiple roots.

c) Calculation of Specified Eigenvectors of a Tridiagonal Matrix

When an accurate eigenvalue of a real symmetric tridiagonal matrix is known the corresponding eigenvector may be computed by inverse iteration and this provides perhaps the most efficient method of determining the eigenvectors corresponding to a relatively small number of selected eigenvalues. When the

eigenvalues are well separated inverse iteration provides an elegant and efficient algorithm.

When eigenvectors corresponding to multiple or very close eigenvalues are required, the determination of fully independent eigenvectors (i.e. of eigenvectors giving full digital information) is quite difficult. The procedure *tristurm* is designed to calculate all the eigenvalues of a symmetric tridiagonal matrix lying in a prescribed interval, and to give the corresponding eigenvectors. The orthogonality of the computed vectors is prescribed by a parameter and to assist in the determination of orthogonal vectors, the organisation of the section dealing with the calculation of the eigenvalues themselves, although based on essentially the same principles as *bisect*, has special features which simplify the determination of orthogonal eigenvectors. When a comparatively small percentage of the eigenvalues and the corresponding eigenvectors are required *tristurm* is the most efficient procedure. The individual eigenvectors provided by *tristurm* are fully as accurate as those given by *jacobi*, *tql 2* or *imtql 2* but the orthogonality of the computed eigenvectors will be less satisfactory with *tristurm*. When we are concerned with eigenvectors associated with clusters of eigenvalues *tristurm* is relatively less efficient and less aesthetically satisfying because of the amount of time spent in the reorthogonalization procedures.

4. Symmetric Band Matrices

Matrices in which the non-zero elements are concentrated in a comparatively narrow band arise quite frequently in practice, usually from finite difference approximations to ordinary or partial differential equations. Generally it is inefficient to treat these by the algorithms recommended for dense matrices, and since it is not uncommon for such matrices to be of high order, storage considerations will often preclude this. In addition to the tridiagonal case, which we have already discussed, three procedures are published for dealing with symmetric band matrices in general.

4.1. Calculation of Eigenvalues

The eigenvalue of smallest modulus of a symmetric band matrix A may usually be computed using procedure *bqr*. (The nature of the QR algorithm is such that no absolute guarantee can be given that the computed eigenvalue really is that of minimum modulus but there is a very high probability that this will be true.) This algorithm may be used to find the eigenvalue nearest to an assigned value t by working with $A - tI$. The procedure is designed so that the m eigenvalues nearest to a given value t may be found by m successive calls of *bqr*, the order of the matrix being diminished after each call. If confirmation is needed that the required eigenvalues have been determined the procedure *bandet 2* may be used, since this gives the number of eigenvalues greater than any assigned value k.

An alternative procedure is *bandrd* which reduces the given matrix to symmetric tridiagonal form. The eigenvalues may then be found by *tql 1*, *imtql 1*,

bisect or *ratqr* as described earlier. The use of *bandrd* therefore gives great flexibility, particularly when combined with *bisect*. *bandrd* is more efficient then *bqr* when any appreciable number of eigenvalues is required.

4.2. Calculation of the Eigenvectors

When eigenvalues of a band matrix have been determined by one of the preceding algorithms the corresponding eigenvectors may be found by inverse iteration using the procedure *symray*. This procedure has the additional refinement of computing improved eigenvalues (using the Rayleigh quotient); these improved values are usually correct to more than single precision.

5. Simultaneous Determination of Dominant Eigenvalues and Eigenvectors of a Symmetric Sparse Matrix

The procedure *ritzit* is designed to find a number of the eigenvalues of maximum modulus and the corresponding eigenvectors of a real symmetric matrix. The matrix A need not be stored explicitly; all that is needed is a procedure for computing Ax from a given vector x. This may result in a remarkable economy of storage. If, for example, A is the matrix arising from a finite difference approximation to a linear partial differential equation with constant coefficients many of its rows will be the same and will contain very few non-zero elements.

By premultiplying with A^{-1} rather than A the eigenvalues of minimum modulus may be determined. More generally by iterating with $(A-pI)^{-1}$ the eigenvalues nearest to p may be determined. In order to take advantage of the sparseness of A it is essential not to compute $(A-pI)^{-1}$ explicitly: $(A-pI)^{-1}x$ is computed by solving $(A-pI)y=x$. Obviously this algorithm can be used when A is a narrow band matrix.

Convergence of *ritzit* for the determination of the m absolutely largest eigenvalues is linear and governed by the separation ratio between the smallest of the considered eigenvalues and the largest remaining one. Therefore, it will often be advantageous to determine more eigenvalues than are wanted by finding a complete cluster rather than trying to separate it.

Clearly the relative effectiveness of the procedure depends on the distribution of the eigenvalues and on the *a priori* knowledge one has of them, since this enables one to choose m efficiently. In practice there are many situations in which this is an extremely efficient algorithm. It is the only algorithm in the series which deals with matrices of an arbitrarily sparse nature.

6. The Generalized Symmetric Eigenvalue Problems $Ax=\lambda Bx$ and $ABx=\lambda x$

A common problem in practice is the determination of the eigensystems of $Ax=\lambda Bx$ and $ABx=\lambda x$ where A and B are symmetric and B is positive definite. Two of the published algorithms are directly relevant in this case.

a) A and B Dense Matrices

The procedures *reduc 1* and *reduc 2* by means of a Cholesky decomposition of B reduce the problems $Ax=\lambda Bx$ and $ABx=\lambda x$ respectively to the standard

eigenproblem for a symmetric matrix P. (Notice that for the case $Ax=\lambda Bx$, the corresponding matrix P is dense even when A and B are narrow band matrices and in the other case the band widths are added.) The eigenvalue problem for P can then be solved by the methods discussed in Section 3, and accordingly we may find all the eigenvalues or selected eigenvalues with or without the corresponding vectors. An eigenvector of the original problem may be determined from those of the matrix P by the procedure *rebak a* or *rebak b*.

b) A and B Band Matrices

When A and B are narrow band matrices the procedure *ritzit* may be used to compute a number of the eigenvalues of largest modulus. To do this an auxiliary procedure is needed for the Cholesky factorization of B into the product LL^T, where L is a lower triangular band matrix. To find the eigenvalues and eigenvectors of $Ax=\lambda Bx$, *ritzit* is used with $L^{-1}AL^{-T}$ as the iteration matrix. Premultiplication of a vector x with $L^{-1}AL^{-T}$ is performed in the steps

$$L^T y=x, \quad Ay=z, \quad Lw=z$$

so that full advantage is taken of the band form of A and L. Similarly for the problem $ABx=\lambda x$, *ritzit* is used with L^TAL as the iteration matrix.

7. Hermitian Matrices

No procedures are included to deal specifically with a complex Hermitian matrix $A+iB$, though complex analogues of the published procedures could be designed. Such problems may be dealt with by working with the real symmetric matrix of order $2n$ defined by

$$\begin{bmatrix} A & -B \\ B & A \end{bmatrix}.$$

If the eigenvalues of the Hermitian matrix are $\lambda_1, \lambda_2, \ldots, \lambda_n$ those of the augmented matrix are $\lambda_1, \lambda_1, \lambda_2, \lambda_2, \ldots, \lambda_n, \lambda_n$. The presence of the double eigenvalues does not lead to a loss of accuracy. If w is one eigenvector of the augmented matrix corresponding to some (double) eigenvalue λ and w^T is partitioned in the form (u^T, v^T) then $u+iv$ is the eigenvector of $A+iB$ corresponding to λ. The other (orthogonal) eigenvector of the augmented matrix turns out to be $(-v^T, u^T)$ and gives up to a factor i the same eigenvector of $A+iB$. If (u^T, v^T) is normalized, so is $u+iv$. The augmented matrix requires twice as much storage as the complex matrix from which it is derived and with efficient coding the analogous complex procedure could usually be about twice as fast as the real procedure applied to the augmented matrix. However, experience suggests that many "complex packages" are so inefficient that it is not uncommon for the real procedure to be the faster.

8. Real Dense Unsymmetric Matrices

It is impossible to design procedures for solving the general unsymmetric eigenproblem which are as satisfactory numerically as those for symmetric (or Hermitian) matrices, since the eigenvalues themselves may be very sensitive to

small changes in the matrix elements. Moreover the matrix may be defective in which case there is no complete set of eigenvectors.

In practice it is virtually impossible to take this into account unless one deals with matrices having integer elements and rational eigenvalues and eigenvectors. With procedures involving rounding errors one cannot demonstrate that a matrix is defective and the published procedures always attempt to find the same number of eigenvectors as eigenvalues. It will often be evident from the computed eigenvectors that the matrix is defective; the corresponding eigenvectors will be almost parallel. It should be emphasized that the computed eigenvalues corresponding to a non-linear elementary divisor will often be by no means close.

When a matrix is defective or close to a defective matrix, it would be more satisfactory to determine invariant subspaces associated with groups of eigenvalues. The most difficult practical problem in designing such a procedure is to determine which eigenvalues to group together. The larger one takes the groups the easier it becomes to determine accurate invariant subspaces, but if the groups are made unnecessarily large then less information is made available. Algorithms of this type are still under development.

The determination of an accurate eigensystem may present practical difficulties if the matrix is badly balanced, that is, if corresponding rows and columns have very different norms. Before using any of the algorithms described in this section it is advisable to balance the matrix using the procedure *balance*. This will usually give a substantial improvement in the accuracy of the computed eigenvalues when the original matrix is badly balanced. If the original matrix is fairly well balanced the use of this procedure will have little effect but since the time taken is quite small compared with that required to find the eigensystem, it is a good discipline to use *balance* in all cases. The procedure *balance* also recognises 'isolated' eigenvalues, that is eigenvalues which are available by inspection without any computation and its use ensures that such eigenvalues are determined exactly *however ill conditioned they may be*.

8.1. Calculation of All the Eigenvalues with or without Eigenvectors

a) The Generalized Jacobi Method

The procedure *eigen* is based on a generalization of the Jacobi method for real symmetric matrices. It seems unlikely that there is a generalization which retains much of the elegance of the Jacobi method but *eigen* is very compact and gives results which, in general, are almost as accurate as can be expected from the sensitivity of the original problem. It is designed to give all the eigenvalues, and to give either the left-hand system of eigenvectors or the right-hand system or neither, as required. The use of eigen is recommended only for matrices expected to have real eigenvalues (see 10.1 (c)).

b) The QR Algorithm

The alternative algorithm for finding all eigenvalues and eigenvectors of a general matrix is based on a combination of a reduction to Hessenberg form

using one of the pairs of procedures *elmhes* and *elmtrans*, *dirhes* and *dirtrans*, or *orthes* and *ortrans* followed by the QR algorithm *hqr 2*. If the procedure *balance* has been used, the eigenvectors of the original matrix are recovered from those of the balanced matrix using the procedure *balbak*. If the eigenvalues only are required then *hqr* is used in place of *hqr 2*.

The choice between *elmhes*, *dirhes* and *orthes* depends on the following considerations. *orthes* has the advantage of leaving the condition numbers of the individual eigenvalues unaltered. It is therefore valuable when one is interested in analysing a matrix rather carefully. It requires twice as much computation as either *dirhes* or *elmhes* and for this reason cannot be regarded as a general purpose procedure.

In *elmhes* considerable advantage can be taken of the presence of zero elements in A, and if A is very sparse *elmhes* is strongly recommended. In *dirhes* inner-products are accumulated whenever possible during the reduction and hence the effect of rounding errors is minimized. The advantage can be appreciable with matrices of high order but on some computers the accumulation of inner-products is so inefficient that the time taken by *dirhes* is prohibitive.

Each eigenvalue λ_i and its corresponding eigenvector v_i is always exact for some matrix $A+E_i$ where $\|E_i\|/\|A\|$ is of the order of magnitude of the machine precision. The residual vector $Av_i-\lambda v_i$ is therefore always small relative to $\|A\|$, independent of its 'accuracy', regarding A as exact. When A has multiple eigenvalues corresponding to linear divisors *hqr 2* determines fully independent vectors. *hqr 2* is very economical on storage.

8.2. The Computation of Selected Eigenvectors

There are no procedures in the series for finding selected eigenvalues, but when all eigenvalues have been computed using *hqr*, eigenvectors corresponding to a small number of selected eigenvalues may be found using the procedure *invit*. As is usual with inverse iteration the speed and accuracy of *invit* are very satisfactory for well separated eigenvalues and the procedure provides an elegant solution to the problem. For coincident and pathologically close eigenvalues the procedure is aesthetically less satisfying. If morethan 25% of the eigenvectors are wanted it is almost invariably more satisfactory to use *hqr 2*.

9. Unsymmetric Band Matrices

The only published procedure dealing specifically with band matrices is *unsray*. This is designed to find accurate eigenvalues and eigenvectors corresponding to a number of approximate eigenvalues. It can accept matrices with a different number of non-zero super-diagonal and sub-diagonal lines. Both left-hand and right-hand eigenvectors are found and the generalised Rayleigh values are computed from them, thereby providing eigenvalues which are usually accurate to more than single precision. It is valuable in connexion with finite difference approximations to partial differential equation eigenvalue problems. The complete problem may be solved with a coarse mesh (ignoring band form) and then selected eigenvalues and eigenvectors determined accurately using a fine mesh.

The method may, of course, be used even if the matrix involved happens to be symmetric; symmetry is in any case ignored in the execution of the algorithm.

10. Dense Unsymmetric Matrices with Complex Elements

10.1. Calculation of All the Eigenvalues with or without Eigenvectors

a) Reduction to Real Case

A complex matrix $A+iB$ may be treated by working with the augmented matrix

$$\begin{bmatrix} A & -B \\ B & A \end{bmatrix}.$$

Corresponding to each *complex* eigenvalue λ_k and eigenvector x_k of the original matrix, the augmented matrix has eigenvalues λ_k and $\bar{\lambda}_k$ and (with exact computation) the corresponding eigenvectors are of the forms

$$\begin{bmatrix} x_k \\ -ix_k \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \bar{x}_k \\ i\bar{x}_k \end{bmatrix}.$$

Corresponding to a *real* eigenvalue λ_k (with eigenvector $x_k = u_k + iv_k$) of the original matrix, the augmented matrix has a double eigenvalue λ_k and independent vectors are given by

$$\begin{bmatrix} u_k \\ v_k \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} -v_k \\ u_k \end{bmatrix}.$$

If the eigenvalues only are computed there is no simple way of telling which of a complex conjugate pair of eigenvalues 'belongs' to the original matrix. The computation of the eigenvectors resolves this difficulty. Partitioning a typical eigenvector of the augmented matrix into u and v we see that corresponding to a complex conjugate pair one of the vectors will be such that $u+iv$ is null (to working accuracy); the associated eigenvalue is the one to be discarded and the vector $u+iv$ derived from the other will be the required vector. Corresponding to a double real eigenvalue the eigenvector $u+iv$ derived from either of the computed eigenvectors gives an eigenvector of the original.

b) The Complex LR Algorithm

There are two procedures *comlr* and *comlr 2* based on the LR algorithm which deal directly with complex matrices. Both depend on a preliminary reduction of the matrix to Hessenberg form using *comhes*. If eigenvalues only are required *comlr* should be used while if both values and vectors are required *comlr 2* is the relevant procedure. For multiple eigenvalues corresponding to linear elementary divisors *comlr 2* gives fully independent eigenvectors. No attempt to detect non-linear elementary divisors is made, but the near parallel nature of the computed eigenvectors usually gives a strong indication. Although one cannot as with *hqr 2* *guarantee* that *comlr 2* will give eigenvalues and eigenvectors with small residuals this is almost certain to be true.

The LR algorithm has been preferred to the complex QR algorithm because of its simplicity; in the real case the QR algorithm was preferred because it deals satisfactorily with complex conjugate eigenvalues.

c) The Generalized Jacobi Method

The procedure *comeig* is based on a generalization of the Jacobi method analogous to that used in *eigen*. Its convergence properties are more satisfactory than those of *eigen* and as regards speed it is to be preferred to the latter for a real matrix with some complex conjugate eigenvalues. It provides both eigenvalues and right-hand eigenvectors. It can be modified in obvious ways to give left-hand eigenvectors or both left-hand and right-hand eigenvectors. For a real matrix *hqr 2* is superior to *comeig* both in speed and storage since *comeig* takes no advantage of the matrix being real. For complex matrices *comeig* is fully comparable with *comlr 2*.

10.2. Calculation of Selected Eigenvectors of a Complex Matrix

There are no procedures for computing selected eigenvalues of a complex matrix. However, when all the eigenvalues have been computed using *comlr*, selected eigenvectors may be found using *cxinvit* and if the number required is small this combination is much more effective. As is usually the case with procedures based on inverse iteration, the algorithm is less pleasing when there are coincident or pathologically close eigenvalues.