

COMPUTING A FEW EIGENVALUES AND
EIGENVECTORS OF A
SYMMETRIC BAND MATRIX

David S. Scott

Department of Computer Sciences
University of Texas at Austin
Austin, Texas 78712

TR-225 April 1983



Computing a Few Eigenvalues and
Eigenvectors of a
Symmetric Band Matrix

David S. Scott
Computer Sciences Department
University of Texas at Austin
Austin, TX 78712

Abstract This paper presents a new combination of the bisection algorithm and the Rayleigh quotient iteration for computing a few eigenvalues of a symmetric band matrix. Both global and local convergence results are proved and numerical examples are presented. The modification of the algorithm needed to handle generalized eigenvalue problems is described.

1. Introduction

This paper describes a new algorithm for computing a few specified eigenvalues and eigenvectors of a symmetric band matrix. The algorithm was developed for use in conjunction with the symmetric block (or band) Lanczos algorithm. At each step of the Lanczos algorithm it is necessary to compute a few eigenvalues (and eigenvectors) of a symmetric band matrix. The eigenvalues are specified by index and an approximate eigenvector is available which is often quite accurate although it can occasionally be completely wrong.

The new algorithm can compute eigenvalues by index or by location (i.e. all the eigenvalues between zero and one) and can take advantage of approximate eigenvectors if they are available. The local and global convergence rates of the algorithm are given. Some numerical tests are described. Finally the changes necessary to make the algorithm applicable to banded generalized eigenvalue problems are described.

2. The Algorithm

The use of the bisection algorithm to compute eigenvalues of tridiagonal matrices was introduced by Givens [1] in 1954. Bisection uses the Sturm sequence property to compute the number of eigenvalues of a symmetric matrix A less than a number σ by computing triangular factorization of $(A - \sigma I)$. Thus given any interval known to contain the desired eigenvalue, it is possible to compute the eigenvalue count at some point in the interval and thereby shrink the size of the interval known to contain the desired eigenvalue. In the absence of any ancillary information the best choice for the shift σ is the midpoint of the interval, which is what the bisection algorithm uses. By repeating the process the eigenvalue can be computed to any desired accuracy up to the roundoff error threshold.

Bisection of tridiagonal matrices is remarkably stable even without pivoting as shown by Wilkinson on page 302 of [3]. A careful implementation of tridiagonal bisection is given on page 249 of [4]. For larger band widths special pivoting is necessary to get a reliable eigenvalue count. An ALGOL code (BANDET2) implementing this pivoting technique is described on page 70 of [4]. For some reason a FORTRAN version of this code has not been incorporated into EISPACK.

Bisection gives a completely reliable way of computing eigenvalues. The eigenvectors can then be obtained by inverse iteration. The main drawbacks with bisection are that its asymptotic convergence rate is only linear and that there is no convenient way to take advantage of approximate eigenvectors. Of course it is possible to switch from bisection to some faster technique once an eigenvalue has been isolated. Wilkinson mentions this on page 306 of [3] although he adds the observation that for tightly clustered eigenvalues the advantage of switching will be slight, since most of the work will be done in the isolation phase.

The basic idea of the new algorithm is to use a faster technique before the desired eigenvalue has been isolated. Instead of just bisecting the current containment interval, it is possible to maintain an approximate eigenvector and use the Rayleigh quotient of the vector as the shift rather than the midpoint of the interval. A step of inverse iteration applied to the approximate eigenvector can be performed at the same time that the latest eigenvalue count is obtained. This is just the Rayleigh quotient iteration (see [2] page 69) except that if the Rayleigh quotient is outside the containment interval the approximate eigenvector should be replaced by a random vector and the bisection shift should be used.

There are two possible reasons why such an algorithm has not been described before. One is the known existence of stagnation points of the Rayleigh quotient iteration. The stagnation points are numerically

unstable, but it is still impossible to prove a global convergence result for an algorithm using Rayleigh quotient shifts. A more practical problem is the fact that the Rayleigh quotient iteration often converges monotonically. All the computed shifts are then on one side of the eigenvalue and the containment interval does not become small. This is fatal if interval widths are used for termination. Furthermore if the desired eigenvalues are specified by index then many steps of the algorithm may be taken before it is discovered to be converging to one of the wrong eigenvalues. This false convergence can significantly degrade the performance of an algorithm based on Rayleigh quotient shifts.

The surprising fact is that both problems can be cured by the same mechanism. It would be best to have a way of choosing the shift so that a desired eigenvalue is always in the smaller subinterval. This is not always possible but it is possible to choose a shift so that the eigenvalue closest to the Rayleigh quotient is in the smaller subinterval. What is needed is a bound on the distance from the Rayleigh quotient to the nearest eigenvalue. Two potentially useful bounds are available. Given a normalized vector x and its Rayleigh quotient θ then there is an eigenvalue λ of A which satisfies (see pages 69 and 222 of [2])

$$|\theta - \lambda| \leq \delta$$

and

$$|\theta - \lambda| \leq \delta^2/\gamma$$

where

$$\delta = \|(A - \theta I)x\|$$

is the residual norm of x and the gap γ is the distance from θ to the next closest eigenvalue of A .

The vector x and its Rayleigh quotient θ are already available from the algorithm and the residual norm δ can be easily computed. Thus the first bound is always computable. The second bound is an improvement over the first whenever $\delta < \gamma$. In general the true gap is not known but a computable lower bound on the gap may be available from the eigenvalue counts obtained from previous factorizations. Provided that θ lies in an interval known to contain only one eigenvalue then a lower bound on γ can be computed as the distance from θ to the nearest endpoint of this interval. We now define γ to be this computable lower bound (if it exists). Thus β , the bound on the distance from θ to the nearest eigenvalue of A , will be just δ unless γ exists and $\delta < \gamma$ in which case $\beta = \delta^2/\gamma$.

If instead of using θ directly as the shift, it is perturbed toward (but not past) the midpoint of the interval by β then the interval known to contain the eigenvalue closest to θ will be shrunk at least by a factor of two and perhaps much more. If the eigenvalue closest to θ is a desired eigenvalue then convergence may proceed at a much faster pace than bisection, while if the eigenvalue closest to θ is not a desired eigenvalue then this fact is discovered immediately and the current approximate eigenvector can be abandoned. If the Rayleigh quotient lies outside the current containment interval then the vector should be replaced by a random vector and the bisection shift should be used. We call this the BPRQS (Bisection with Perturbed Rayleigh Quotient Shift) algorithm. A formal description of the algorithm follows (for finding one eigenvalue and eigenvector) is as follows:

BPRQS Algorithm

Given an interval $[\kappa_1, \zeta_1]$ known to contain the desired eigenvalue, a normalized vector x_1 , θ_1 the Rayleigh quotient of x_1 , δ_1 the residual norm of x_1 , and γ_1 the computable lower bound on the gap between θ_1 and the second closest eigenvalue of A (if it exists):

For $j = 1, 2, \dots$ until convergence do 1-4

1. Compute the minimum error bound β_j :

If (γ_j does not exist or $\gamma_j < \delta_j$)

Then $\beta_j = \delta_j$

Else $\beta_j = \delta_j^2 / \gamma_j$

2. Select a shift σ_j :

Let $\mu_j = (\kappa_j + \zeta_j) / 2$.

If $\theta_j < \kappa_j$ or $\theta_j > \zeta_j$

Then $\sigma_j = \mu_j$ and set x_j to a random vector

Else If $\theta_j < \mu_j$

Then $\sigma_j = \min\{\mu_j, \theta_j + \beta_j\}$

Else $\sigma_j = \max\{\mu_j, \theta_j - \beta_j\}$

3. Factor $(A - \sigma_j I)$ to get the new eigenvalue count, solve

$(A - \sigma_j I)y_{j+1} = x_j$, and normalize y_{j+1} to get x_{j+1}

4. Update the values:

If the desired eigenvalue is smaller than σ_j

Then $\kappa_{j+1} = \kappa_j$ and

$\zeta_{j+1} = \sigma_j$

Else $\kappa_{j+1} = \sigma_j$ and

$\zeta_{j+1} = \zeta_j$

$\theta_{j+1} = x_{j+1}^T A x_{j+1}$

$\delta_{j+1} = \|(A - \theta_{j+1} I)x_{j+1}\|$

Update γ (if possible) from θ_{j+1} and the previous

eigenvalue counts.

Termination is based on the width of the containment interval or on the error bound once the

computed gap is larger than the residual norm.

3. Global Convergence

The BPRQS algorithm converges globally to a particular eigenvalue with an asymptotic convergence rate which is at least linear with rate constant at least $\sqrt{1/2}$, as shown by the following theorem.

Theorem 1: For all steps j of the BPRQS algorithm, the containment interval after two more steps, that is $[\kappa_{j+2}, \zeta_{j+2}]$, will satisfy at least one of:

1. $\zeta_{j+2} - \kappa_{j+2} \leq (\zeta_j - \kappa_j)/2$
2. $[\kappa_{j+2}, \zeta_{j+2}]$ contains fewer eigenvalues than $[\kappa_j, \zeta_j]$.

Proof: If θ_j is not in $[\kappa_j, \zeta_j]$ then the first step is a bisection step and the theorem is true. Without loss of generality we now assume that θ_j is in $[\kappa_j, \mu_j]$.

If $\theta_j + \beta_j > \mu_j$ then the next step is a bisection step and the theorem is true. So we may assume that the next shift will be $\sigma_j = \theta_j + \beta_j$.

If the desired eigenvalue is in $[\kappa_j, \sigma_j]$ then the theorem is true. So we may assume that $\kappa_{j+1} = \sigma_j$ and $\zeta_{j+1} = \zeta_j$.

If θ_{j+1} is not in $[\kappa_{j+1}, \zeta_{j+1}]$ then the next step will be a bisection step and the theorem will be true. So we may assume that θ_{j+1} is in $[\kappa_{j+1}, \zeta_{j+1}]$.

Assume for now that $\theta_{j+1} \leq \mu_{j+1}$.

If $\theta_{j+1} + \beta_{j+1} \geq \mu_{j+1}$ then the next step will be a bisection step and the theorem is true. So we may assume that $\sigma_{j+1} = \theta_{j+1} + \beta_{j+1}$.

If the desired eigenvalue is in $[\kappa_{j+1}, \sigma_{j+1}]$ then the theorem is true. So we may assume that $\kappa_{j+2} = \sigma_{j+1}$ and $\zeta_{j+2} = \zeta_{j+1}$.

If there is any eigenvalue in $[\kappa_j, \kappa_{j+2}]$ then the theorem is true. The bound β is computed using γ only if the interval $[\theta - \delta, \theta + \delta]$ is known to contain only the desired eigenvalue. In this case $[\theta - \beta, \theta + \beta]$ also contains only the desired eigenvalue. Since we have assumed that the desired eigenvalue is not in this interval at both step j and step $j+1$ it follows that both $\beta_j = \delta_j$ and $\beta_{j+1} = \delta_{j+1}$.

The following lemma is needed to prove under the above assumptions that there is an eigenvalue in the interval $[\kappa_j, \kappa_{j+2}]$.

Lemma 2: Let θ be the Rayleigh quotient of the normalized vector x and let δ be its residual norm ($= \|(A - \theta I)x\|$). Let $\mu y = (A - \sigma I)^{-1}x$ for some number σ , where μ ($= 1/\|(A - \sigma I)^{-1}x\|$) is chosen to normalize y , let α be the Rayleigh quotient of y , and let ρ be its residual norm. Then if $\theta \leq \sigma \leq \alpha$ then

$$\rho \leq \delta^1$$

and

$$\alpha - \sigma \leq \delta.$$

¹This generalizes Kahan's monotone residual theorem for the Rayleigh quotient iteration. See p. 75 of [2].

Proof:

$$\|(A - \theta I)x\| \geq y^T(A - \theta I)x$$

by the Cauchy-Schwarz inequality,

$$\begin{aligned} &= y^T(A - \sigma I)x + (\sigma - \theta)y^T x \\ &= y^T(A - \sigma I)x + \mu(\sigma - \theta)y^T(A - \sigma I)y \\ &= y^T(A - \sigma I)x + \mu(\sigma - \theta)(\alpha - \sigma) \\ &\geq y^T(A - \sigma I)x \end{aligned}$$

by hypothesis,

$$= \|(A - \sigma I)y\|$$

since x is parallel to $(A - \sigma I)y$,

$$\geq \|(A - \alpha I)y\|$$

since the Rayleigh quotient minimizes the residual norm.

This proves the first result. To get the second we apply the Cauchy-Schwarz inequality to the next to last line above to get

$$\begin{aligned} &\geq y^T(A - \sigma I)y \\ &= \alpha - \sigma. \end{aligned}$$

By elementary linear algebra there is an eigenvalue in $[\theta_{j+1} - \delta_{j+1}, \theta_{j+1} + \delta_{j+1}]$. $\beta_{j+1} = \delta_{j+1}$ as shown above, $\delta_{j+1} \leq \delta_j$ by the lemma, $\sigma_j = \theta_j + \beta_j$ by construction, and $\theta_j \geq \kappa_j$. Therefore

$$\begin{aligned} [\kappa_j, \kappa_{j+2}] &= [\kappa_j, \theta_{j+1} + \beta_{j+1}] \\ &= [\kappa_j, \theta_{j+1} + \delta_{j+1}] \\ &\supseteq [\theta_{j+1} - \delta_{j+1}, \theta_{j+1} + \delta_{j+1}] \end{aligned}$$

and hence contains an eigenvalue.

Finally assume $\theta_{j+1} \geq \mu_{j+1}$. The new containment interval will be smaller than $[\sigma_j, \theta_{j+1}]$ and by the lemma, $\theta_{j+1} - \sigma_j \leq \delta_j$ and $\delta_j \leq (\kappa_j - \zeta_j)/2$ by assumption (since the first shift was less than a bisection shift). Therefore $\zeta_{j+2} - \kappa_{j+2} \leq (\zeta_j - \kappa_j)/2$ and the theorem is true.

This completes the proof.

4. Local Convergence

The previous section showed that the BPRQS algorithm converges globally with asymptotic rate of convergence no worse than linear with rate constant $\text{SQRT}(1/2)$. With a careful analysis it is possible to improve the rate constant to $1/2$ but impossible to prove that the asymptotic convergence rate is better than linear. The reason for this is that the global convergence result only guarantees convergence to the eigenvalue by the bisection process, it does not guarantee convergence to the eigenvector. This convergence failure can happen only if the starting vector x_0 is orthogonal to the desired eigenvector. Furthermore during the algorithm it is very likely that the current vector would be periodically discarded and a new random vector generated. To prevent convergence to the desired eigenvector each of these subsequent vectors would also have to be orthogonal to the desired eigenvector. This event has probability zero. We now assume that the vectors x_j converge to the desired eigenvector. With this assumption it is possible to prove that the convergence of the BPRQS algorithm is asymptotically quadratic or cubic.

Theorem 3: Assume that the sequence of vectors x_j generated by the BPRQS algorithm converges to the desired eigenvector z . Let λ be the eigenvalue associated with z . If λ is a simple eigenvalue then the convergence is asymptotically cubic. Otherwise the convergence is asymptotically quadratic.

Proof: Assume that the projection of the vector x_j onto the eigenspace associated with λ is the vector z and assume that

$$x_j = \nu z + \epsilon w$$

where w and z are unit vectors, w is orthogonal to z , and $\nu = \text{SQRT}(1 - \epsilon^2)$. The Rayleigh quotient of x_j is

$$\theta_j = \lambda + \epsilon^2 (w^T A w - \lambda)$$

which is obviously $\lambda + O(\epsilon^2)$.

The residual norm of x_j is

$$\|(A - (\lambda + O(\epsilon^2))I)x_j\| = \epsilon \|(A - \lambda I)w\| + O(\epsilon^2)$$

If λ is a multiple eigenvalue then the next shift σ_j will be within $O(\epsilon)$ of λ and the next vector will be

$$y_{j+1} = (A - (\lambda + O(\epsilon))I)^{-1} x_j.$$

Thus the z component of x_j will be amplified by $1/\epsilon$ while the other components will be amplified by at most $1/\gamma$ where γ is the gap between λ and the distinct eigenvalue closest to λ . If $\epsilon \ll \gamma$ then after normalization

$$x_{j+1} = z + O(\epsilon^2).$$

If λ is a simple eigenvalue then eventually the minimum error bound will be δ_j^2/γ_j and thus the next shift σ_j will be within $O(\epsilon^2)$ of λ . The next vector y_{j+1} will have its z component amplified by $O(1/\epsilon^2)$ while the others remain bounded so that after normalization

$$x_{j+1} = z + O(\epsilon^3)$$

and the theorem is proved.

5. Numerical Results

The implementation of the algorithm used in this section computes the desired eigenvalues sequentially from left to right. Containment intervals for all the desired eigenvalues are updated after each factorization. Vectors discarded in the course of computing one eigenvalue are saved if appropriate for

computing later eigenvalues.

This section gives the results of a few numerical experiments. In every test except the last, the matrix A was of the form HDH where $H = I - 2ww^T/w^T w$ was a Householder reflection matrix with a random vector w and D was a diagonal matrix containing the eigenvalues. This form allows the easy specification of the eigenvalues without suffering from the special rounding error characteristics of using a diagonal matrix directly. Of course the resulting matrix is not banded but $(A - \sigma I)^{-1}$ can be computed as $H(D - \sigma I)^{-1}H$ which keeps the the cost of the tests down. The tests were run on a DEC-20 computer in double precision.

The first example had $D = \text{diag}(1, 2, 3, \dots, 50)$. The smallest eigenvalue was computed from an initial containment interval $[-6, 55]$ and a random vector. The behavior of the algorithm is given in Table 1.

Table 5-1: Behavior of BPRQS Algorithm

step	RQ	Residual	Error Bound	Shift
1	27.204	13.50	13.50	25.5000
2	25.715	2.09	2.09	10.2500 *
3	10.096	1.87	1.87	8.2220
4	9.232	.837	.837	1.6611 *
5	2.825	3.11	3.11	-1.6945
6	4.756	7.00	7.00	-.0417 *
7	3.267	4.80	4.80	.7846 *
8	3.964	5.78	5.78	1.1978 *
9	1.319	1.24	1.24	.9912
10	1.000	.0555	.00505	1.00534
11	1.000	.515e-4	.434e-8	.9999999972
12	1.000	.962e-13	.304e-15	terminate

* indicates a bisection shift and randomization of the eigenvector approximation

As can be seen from Table 1, it takes a while for the shift to get nearer to 1 than to any other eigenvalue, at which point the cubic rate of convergence of the algorithm becomes apparent.

It is more efficient in terms of factorizations per eigenvalue to compute several adjacent eigenvalues since the information gained while locating the first eigenvalue gives a head start on finding the rest. Table 2 shows the number of factorizations needed to compute the first ten eigenvalues of the same matrix A using the same initial containment interval.

To test the effects of multiple eigenvalues on the algorithm the second test matrix had the same eigenvalues as the first except that five extra eigenvalues were added to make the eigenvalues 1, 2, 3, 4, and 5 double. The behavior of the algorithm on computing the smallest eigenvalue of the matrix, starting with the interval $[-5, 56]$ and a random vector, is shown in table 3. The quadratic convergence can be seen easily as well as the additional penalty for terminating due to interval width rather than due to small residual.

As before in the simple eigenvalue case, the number of factorizations per eigenvalue was reduced if several eigenvalues are computed at once. The number of factorizations needed for each of the ten smallest eigenvalues for the second test matrix are given in table 4. Each second eigenvalue was already known to the desired accuracy, the factorization was needed only in computing the corresponding eigenvector. The total number of factorizations needed to compute the ten eigenvalues was 54. This

Table 5-2: Number of Factorizations Per Eigenvalue

Eigenvalue	Number of Factorizations
1.	11
2.	8
3.	6
4.	4
5.	7
6.	3
7.	4
8.	5
9.	4
10.	3

Table 5-3: Behavior of BPRQS Algorithm on Multiple Eigenvalues

step	RQ	Residual	Error Bound	Shift
1	19.957	14.9	14.9	25.500 *
2	25.098	4.03	4.03	21.067
3	21.048	.514	.514	20.535
4	20.998	.075	.075	7.767 *
5	8.264	4.73	4.73	1.384 *
6	1.685	1.96	1.96	-1.808 *
7	3.972	6.05	6.05	-.212 *
8	3.464	4.68	4.68	.586 *
9	1.201	1.39	1.39	.985 *
10	1.000	.498e-2	.498e-2	1.005
11	1.000	.227e-4	.227e-4	.999977
12	1.000	.511e-9	.511e-9	1.0000000005107
13	1.000	.334e-15	.334e-15	.99999999999999666
14	1.000	.334e-15	.334e-15	1.00000000000000334
15	1.000	termination due to interval width		

* indicates a bisection shift and randomization
of the eigenvector approximation

Table 5-4: Number of Factorizations Per Eigenvalue

Eigenvalue	Number of Factorizations
1.0	14
1.0	1
2.0	10
2.0	1
3.0	9
3.0	1
4.0	7
4.0	1
5.0	9
5.0	1

compares to the 55 factorizations needed to compute all the eigenvalues in table 2.

The behavior of the algorithm on close but not multiple eigenvalues is similar. The third test matrix had eigenvalues i^{-3} , $i = 1, 2, \dots, 50$. The code took 60 factorizations to compute the ten smallest eigenvalues which are bunched closely together near zero. When asked to compute the ten largest eigenvalues the code needed 61 factorizations.

The final test matrix used was one of the tridiagonal matrices used to test the tridiagonal bisection code on p. 255 of [4]. The off diagonal elements are all ones while the diagonal is $\text{diag}(100, 90, 80, \dots, 10, 0, 10, \dots, 100)$. All twenty one eigenvalues were computed to fifteen figures of accuracy using 93 factorizations starting with random vectors and the containment interval $[-1, 101]$, which is known to contain all the eigenvalues by the Gerschgorin circle theorem. The bisection algorithm took 345 factorizations to compute the eigenvalues to seven figures of accuracy.

6. Generalized Eigenvalue Problems

Both the bisection algorithm and the Rayleigh quotient iteration can be applied to the generalized eigenvalue problem

$$(A - \lambda M)z = 0$$

provided that both A and M are symmetric and M is positive definite. The only problem is that given an approximate eigenvector x with $x^T M x = 1$ and its Rayleigh quotient $\theta = x^T A x$, the residual norm needed by the algorithm is the M^{-1} norm of $(A - \theta M)x$. Computing an M^{-1} norm requires the factorization of M . However neither the time needed to compute this factorization nor the space needed to store it make a significant increase in the time and space needed for the algorithm in the standard case. Once the factorization of M is available the algorithm proceeds as described above. Thus the algorithm is equally applicable to generalized eigenvalue problems.

7. Conclusions

This paper has described and analyzed a new combination of the Rayleigh quotient iteration and the bisection algorithm and has shown that it is an effective way to compute a few eigenvalues and eigenvectors of a symmetric band matrix or a symmetric definite banded generalized eigenvalue problem.

8. Acknowledgment

The author wishes to thank the referees for their many helpful comments.

References

- [1] Givens, W.
Numerical Computation of Characteristic Values of a Real Symmetric Matrix.
Technical Report ORNL-1574, Oak Ridge National Laboratory, 1954.
- [2] Parlett, B. N.
The Symmetric Eigenvalue Problem.
Prentice-Hall, 1980.
- [3] Wilkinson, J. H.
The Algebraic Eigenvalue Problem.
Oxford University Press, 1965.
- [4] Wilkinson, J. H., and C. Reinsch.
Handbook for Automatic Computation.
Springer Verlag, 1971.