Numerische Mathematik 9, 386–393 (1967)

Handbook Series Linear Algebra

Calculation of the Eigenvalues of a Symmetric Tridiagonal Matrix by the Method of Bisection*

Contributed by
W. BARTH, R. S. MARTIN and J. H. WILKINSON

Theoretical Background

The procedure *bisect* is designed to replace the procedures *tridibi 1* and *2* given in [5]. All three procedures are based essentially on the following theorem.

Given a symmetric tridiagonal matrix with diagonal elements c_1 to c_n and off-diagonal elements b_2 to b_n (with b_1 set equal to zero for completeness) and $b_i \neq 0$, then for any number λ let the sequence $p_0(\lambda), \ldots, p_n(\lambda)$ be defined by

$$p_0(\lambda) = 1, \qquad p_1(\lambda) = c_1 - \lambda, \tag{1}$$

$$p_i(\lambda) = (c_i - \lambda)\, p_{i-1}(\lambda) - b_i^2 p_{i-2}(\lambda) \qquad (i = 2, \ldots, n). \tag{2}$$

Then, in general, the number, $a(\lambda)$, of disagreements in sign between consecutive numbers of the sequence is equal to the number of eigenvalues smaller than λ.

When carried out in floating-point arithmetic, this method is extremely stable but unfortunately, even for matrices of quite modest order, it is common for the later $p_i(\lambda)$ to pass outside the range of permissible numbers. In practice both underflow and overflow occur and it is almost impossible to scale the matrix to avoid this. The difficulty is particularly acute when there are a number of very close eigenvalues since $p_n(\lambda) = \Pi(\lambda - \lambda_i)$ is then very small for any λ in their neighbourhood. The zeros of each $p_r(\lambda)$ separate those of $p_{r+1}(\lambda)$ and accordingly quite a number of the $p_i(\lambda)$ other than $p_n(\lambda)$ may also be very small and give rise to underflow.

The difficulty is avoided by replacing the sequence of $p_i(\lambda)$ by a sequence of $q_i(\lambda)$ defined by

$$q_i(\lambda) = p_i(\lambda)/p_{i-1}(\lambda) \qquad (i = 1, \ldots, n) \tag{3}$$

$a(\lambda)$ is now given by the number of negative $q_i(\lambda)$. The $q_i(\lambda)$ satisfy the relations

$$q_1(\lambda) = c_1 - \lambda, \tag{4}$$

$$q_i(\lambda) = (c_i - \lambda) - b_i^2/q_{i-1}(\lambda) \qquad (i = 2, \ldots, n). \tag{5}$$

* *Editor's note.* In this fascicle, prepublication of algorithms from the Linear Algebra series of the Handbook for Automatic Computation is continued. Algorithms are published in ALGOL 60 reference language as approved by the IFIP. Contributions in this series should by styled after the most recently published ones. Inquiries are to be directed to the editor. F. L. BAUER, Munich

At first sight these relations look very dangerous, since it is possible for $q_{i-1}(\lambda)$ to be zero for some i; but in such cases it is merely necessary to replace the zero $q_{i-1}(\lambda)$ by a suitably small quantity and the error analysis for the $p_i(\lambda)$ sequence [4, 6] applies almost unaltered to the $q_i(\lambda)$. The $q_i(\lambda)$ do not suffer from the problems of underflow and overflow associated with the $p_i(\lambda)$. Comparing the computation of $q_i(\lambda)$ with that of $p_i(\lambda)$ we observe that two multiplications have been replaced by one division; further we have only to detect the sign of each $q_i(\lambda)$ instead of comparing the signs of $p_{i-1}(\lambda)$ and $p_i(\lambda)$. It is easy to see that when working with the $q_i(\lambda)$ the condition $b_i \neq 0$ can be omitted.

In the earlier paper [5] each eigenvalue was found independently by bisection. This is inefficient since, as observed by GIVENS in his original paper [2], some of the information obtained when determining one eigenvalue is, in general, of significance in the determination of other eigenvalues. In *bisect* full advantage is taken of all relevant information and this results in a very substantial saving of time when the matrix has a number of close or coincident eigenvalues.

Applicability

bisect may be used to find the eigenvalues $\lambda_{m1}, \lambda_{m1+1}, \ldots, \lambda_{m2}$ $(\lambda_{i+1} \geqq \lambda_i)$ of a symmetric tridiagonal matrix of order n. Unsymmetric tridiagonal matrices A with $a_{i,i+1}=f_i$ and $a_{i+1,i}=g_i$ may be treated provided $f_i g_i \geqq 0$ $(i=1, \ldots, n-1)$ by taking $b_{i+1}^2=f_i g_i$.

Formal Parameter List

Input to procedure *bisect*

c — an $n \times 1$ array giving the diagonal elements of a tridiagonal matrix.

b — an $n \times 1$ array giving the sub-diagonal elements. *b*[1] may be arbitrary but is replaced by zero in the procedure.

beta — an $n \times 1$ array giving the squares of the sub-diagonal elements. *beta*[1] may be arbitrary but is replaced by zero in the procedure. Both *b*[*i*] and *beta*[*i*] are given since the squares may be the primary data. If storage economy is important then one of these can be dispensed with in an obvious way.

n — the order of the tridiagonal matrix.

m1, m2 — the eigenvalues $\lambda_{m1}, \lambda_{m1+1}, \ldots, \lambda_{m2}$ are calculated (λ_1 is the smallest eigenvalue). $m1 \leqq m2$ must hold otherwise no eigenvalues are computed.

eps1 — a quantity affecting the precision to which the eigenvalues are computed (see "Discussion of numerical properties").

relfeh — the smallest number for which $1+relfeh>1$ on the computer.

Output of procedure *bisect*

eps2 — gives information concerning the accuracy of the results (see "Discussion of numerical properties").

z — total number of bisections to find all required eigenvalues.

x — array *x*[*m1*:*m2*] contains the computed eigenvalues.

ALGOL Programs

```
procedure bisect (c, b, beta, n, m1, m2, eps1, relfeh) res:(eps2, z, x);
value n, m1, m2, eps1, relfeh;
real eps1, eps2, relfeh; integer n, m1, m2, z; array c, b, x, beta;
comment c is the diagonal, b the sub-diagonal and beta the squared subdiagonal
        of a symmetric tridiagonal matrix of order n. The eigenvalues
        lambda[m1], ..., lambda[m2], where m2 is not less than m1 and
        lambda[i+1] is not less than lambda[i], are calculated by the method
        of bisection and stored in the vector x. Bisection is continued until the
        upper and lower bounds for an eigenvalue differ by less than eps1,
        unless at some earlier stage, the upper and lower bounds differ only
        in the least significant digits. eps2 gives an extreme upper bound for
        the error in any eigenvalue, but for certain types of matrices the small
        eigenvalues are determined to a very much higher accuracy. In this
        case, eps1 should be set equal to the error to be tolerated in the smallest
        eigenvalue. It must not be set equal to zero;
begin   real h, xmin, xmax; integer i;
        comment Calculation of xmin, xmax;
        beta[1] := b[1] := 0;
        xmin := c[n] - abs(b[n]);
        xmax := c[n] + abs(b[n]);
        for i := n - 1 step -1 until 1 do
        begin h := abs(b[i]) + abs(b[i+1]);
              if c[i] + h > xmax then xmax := c[i] + h;
              if c[i] - h < xmin then xmin := c[i] - h;
        end i;
        eps2 := relfeh × (if xmin + xmax > 0 then xmax else - xmin);
        if eps1 ≦ 0 then eps1 := eps2;
        eps2 := 0.5 × eps1 + 7 × eps2;
        comment Inner block;
        begin   integer a, k; real q, x1, xu, x0; array wu[m1:m2];
                x0 := xmax;
                for i := m1 step 1 until m2 do
                begin x[i] := xmax; wu[i] := xmin
                end i;
                z := 0;
                comment Loop for the k-th eigenvalue;
                for k := m2 step -1 until m1 do
                begin xu := xmin;
                      for i := k step -1 until m1 do
                      begin if xu < wu[i] then
                            begin xu := wu[i]; go to contin
                            end
                      end i;
                contin: if x0 > x[k] then x0 := x[k];
                      for x1 := (xu + x0)/2 while x0 - xu > 2 × relfeh ×
                                                  (abs(xu) + abs(x0)) + eps1 do
```

```
                begin z := z + 1;
                    comment Sturms sequence;
                    a := 0; q := 1;
                    for i := 1 step 1 until n do
                    begin
                        q := c[i] - x1 - (if q ≠ 0 then beta[i]/q
                                          else abs(b[i])/relfeh);
                        if q < 0 then a := a + 1
                    end i;
                    if a < k then
                    begin if a < m1 then
                            xu := wu[m1] := x1
                            else
                            begin xu := wu[a + 1] := x1;
                                  if x[a] > x1 then x[a] := x1
                            end
                    end
                    else x0 := x1
                end x1;
                x[k] := (x0 + xu)/2
            end k
        end inner block
end bisect;
```

Organisational and Notational Details

The value of *relfeh* is machine dependent and is directly related to the precision of the arithmetic which is used. On a computer with a t digit binary mantissa *relfeh* is of the order of 2^{-t}. Errors in c_i and b_i of up to $|c_i|$ *relfeh* and $|b_i|$ *relfeh* in absolute magnitude may be introduced by the digital representation.

From GERSCHGORINS' theorem the eigenvalues are all contained in the union of the n intervals $c_i \pm (|b_i| + |b_{i+1}|)$ with $b_1 = b_{n+1} = 0$. Hence *xmax* and *x min* defined by

$$\begin{matrix} xmax \\ xmin \end{matrix} = \begin{matrix} max \\ min \end{matrix} \{c_i \pm (|b_i| + |b_{i+1}|)\} \qquad i = 1, 2, \ldots, n \tag{6}$$

can be used as initial upper and lower bounds for the eigenvalues.

When computing a typical eigenvalue λ_k two quantities *x0* and *xu* which are current upper and lower bounds for λ_k are stored. Bisection is continued as long as

$$x0 - xu > 2\,relfeh(|xu| + |x0|) + eps1 \tag{7}$$

where *eps 1* is a preassigned tolerance. The significance of this criterion is discussed in the next section. When (7) is no longer satisfied $\frac{1}{2}(x0 + xu)$ gives the current eigenvalue λ_k.

If at any stage in the computation of a Sturm sequence $q_{i-1}(\lambda)$ is zero then (5) is replaced by

$$q_i(\lambda) = (c_i - \lambda) - |b_i|/relfeh. \tag{8}$$

This means in effect that $q_{i-1}(\lambda)$ is replaced by $|b_i|$ *relfeh* (note that this is positive; this is essential because the zero $q_{i-1}(\lambda)$ is treated as positive) and this is equivalent to replacing c_{i-1} by $c_{i-1}+|b_i|$ *relfeh*.

The eigenvalues are found in the order $\lambda_{m2}, \lambda_{m2-1}, \ldots, \lambda_{m1}$. Two arrays *wu*[*m1*:*m2*] and *x*[*m1*:*m2*] are used to store relevant information on the lower and upper bounds of the required eigenvalues. Initially we have $xu = wu[i] = xmin$, $x0 = x[i] = xmax$ $(i = m1, \ldots, m2)$. In practice it is not necessary to keep all the *wu*[*i*] and *x*[*i*] fully updated provided it is possible to determine the current best upper and lower bounds for any eigenvalue when required. This is achieved as follows.

If, during the calculation of λ_k corresponding to an argument *x1*, we have $a(x1) \geqq k$, the only useful deduction is $x0 := x1$. If, however, $a(x1) < k$, then *x1* is an upper bound for $\lambda_{m1}, \lambda_{m1+1}, \ldots, \lambda_a$ and a lower bound for $\lambda_{a+1}, \lambda_{a+2}, \ldots, \lambda_k$. Hence in any case $xu := x1$. If $a < m1$ then $wu[m1] := x1$; otherwise $wu[a+1] := x1$ and $x[a] := x1$ if this is an improved upper bound.

To find the initial values of *x0* and *xu* when computing λ_k we take *xu* to be the largest of *xmin* and *wu*[*i*] $(i = m1, \ldots, k)$, and *x0* to be the smaller of *x0* and *x*[*k*]. No superfluous information is stored and the total number of bisections required when there are multiple or close eigenvalues is significantly less than when the eigenvalues are well separated.

Discussion of Numerical Properties

To understand the criterion for the termination of the bisection process it is necessary to consider the limitations on the accuracy obtainable by the bisection method. *In general*, using floating point computation with a t digit mantissa, errors of the order of magnitude $2^{-t}\, max[|xmax|, |xmin|]$ are inevitable in all eigenvalues and they cannot be reduced by increasing the number of bisection steps. This means that, in general, the relative error in the smaller eigenvalues is higher than in larger ones.

However, there are certain types of matrices for which it is possible to determine the lower eigenvalues with the same low relative error as the higher eigenvalues. An example of such a matrix is given by

$$c_i = i^4, \qquad b_i = i - 1 \qquad (i = 1, \ldots, 30). \tag{9}$$

This matrix has eigenvalues which are roughly of the order of magnitude i^4 $(i = 1, \ldots, 30)$ and variations of the elements by up to one part in 2^t change all the eigenvalues by roughly one part in 2^t. The smaller eigenvalues are therefore determined by the data to the same relative accuracy as the larger ones. Moreover by continuing the bisection process long enough this accuracy can actually be attained in practice though it may be more efficient to use the QD-algorithm [3] or the symmetric QR algorithm [1]. Matrices of this type are common in theoretical physics where they arise by truncating infinite tridiagonal matrices. Here it is the smallest eigenvalues of the infinite matrix which are required and by taking a truncated matrix of sufficiently high order these can be obtained to any prescribed accuracy.

The criterion (7) enables us to deal with matrices of the non-special or special types with considerable flexibility. If we have a non-special matrix, then a value of *eps*1 approximately equal to $2^{-t}\, max[|xmax|, |xmin|]$ will give the maxi-

mum attainable accuracy in each eigenvalues without any superfluous bisection steps. If a lower accuracy is adequate then this is achieved by taking a larger value of *eps 1*; for example if eigenvalues are required up to the third decimal place (*not the third significant figure*), *eps 1* may be taken equal to $\frac{1}{2}10^{-3}$ and if the norm of the matrix is of order unity the term $2\,relfeh\times(|xu|+|x0|)$ on the right hand side of (7) will be of negligible significance.

With the special type of matrix on the other hand, when the small eigenvalues are required with low relative error it is the term $2\,relfeh(|xu|+|x0|)$ which plays the dominant role. If the term *eps1* is omitted, then, in general, bisection would continue until all but the last one or two digits of *x0* and *xu* agree. However, if one of the eigenvalues were zero, this condition might never be attained and accordingly *eps1* should be set equal to the error which is acceptable in the smallest eigenvalue. To guard against the accidental assignment of a non-positive value of *eps1*, such values are replaced by $relfeh\times max[|xmax|, |xmin|]$, thus avoiding the danger of indefinite cycling.

Bounds for the errors may be obtained by the method described in [*4, 6*]. The overall bound for the error in any eigenvalue is given by

$$\tfrac{1}{2}\,eps1 + 7\,relfeh\times max[|x\,max|, |x\,min|] \tag{10}$$

and this quantity is computed and output as *eps 2*. It should be appreciated that for the matrices of special type the errors in the smaller eigenvalues will be much smaller than this.

Test Results

The procedure *bisect* has been tested on a number of matrices using the IBM 7040 at Darmstadt and the KDF 9 at the National Physical Laboratory. Tests of an earlier version of the procedure were also carried out on the CDC 1604 computer of the Swiss Federal Institute of Technology, Zurich, Switzerland. The following illustrate the points of interest discussed earlier.

The first tridiagonal matrix was obtained by the Householder reduction of a matrix A of order 50 with $a_{ij}=1$ (all i, j). This matrix has the elements

$$\begin{aligned} &c_1=1, \quad c_2=49, \quad c_i=0 \qquad (i=3,\ldots,50),\\ &b_2=7, \quad b_i=0 \qquad\qquad\quad (i=3,\ldots,50), \end{aligned}$$

and has one eigenvalue equal to 50 and the other 49 equal to zero. This matrix gives rise to underflow with the original procedures *tridibi 1* and *2* [*5*]. The computed eigenvalues on KDF 9 ($relfeh=2^{-39}$) and with $eps1=10^{-10}$ were

$$\lambda_1, \lambda_2, \ldots, \lambda_{49}=2.27373\,675443_{10}-11, \quad \lambda_{50}=5.00000\,000001_{10}+1$$

and the output value of *eps2* was $7.6304_{10}-10$. A total of 73 bisections were needed, none being required for the eigenvalues $\lambda_1, \ldots, \lambda_{48}$. The number of iterations needed with the earlier procedures would have been 50×39.

The second matrix is of the special type and is of order 30 with the elements

$$c_i=i^4, \qquad b_i=i-1 \qquad (i=1, 2, \ldots, 30).$$

This was solved on KDF 9 using $eps1=10^{-8}$, 10^{-10}, and 10^{-12} respectively. The results obtained are given in Table 1.

Table 1

$eps1 = 10^{-8}$ eigenvalues		$eps1 = 10^{-10}$ eigenvalues	$eps1 = 10^{-12}$ eigenvalues
$9.3340\,7085\,678_{10}-1$	$6.5536\,0008\,557_{10}+4$	$9.3340\,7084\,825_{10}-1$	$9.3340\,7084\,869_{10}-1$
$1.6005\,0653\,722_{10}+1$	$8.3521\,0007\,641_{10}+4$	$1.6005\,0653\,704_{10}+1$	$1.6005\,0653\,703_{10}+1$
$8.1010\,1005\,468_{10}+1$	$1.0497\,6000\,687_{10}+5$	$8.1010\,1005\,456_{10}+1$	The remaining
$2.5600\,8066\,892_{10}+2$	$1.3032\,1000\,620_{10}+5$	$2.5600\,8066\,893_{10}+2$	values as for
$6.2500\,6102\,376_{10}+2$	$1.6000\,0000\,563_{10}+5$	$6.2500\,6102\,372_{10}+2$	$eps1 = 10^{-10}$
$1.2960\,0467\,858_{10}+3$	$1.9448\,1000\,514_{10}+5$	$1.2960\,0467\,858_{10}+3$	
$2.4010\,0367\,061_{10}+3$	$2.3425\,6000\,470_{10}+5$	$2.4010\,0367\,062_{10}+3$	
$4.0960\,0294\,507_{10}+3$	$2.7984\,1000\,431_{10}+5$	The remaining values	
$6.5610\,0241\,013_{10}+3$	$3.3177\,6000\,399_{10}+5$	as for $eps1 = 10^{-8}$	
$1.0000\,0020\,063_{10}+4$	$3.9062\,5000\,369_{10}+5$		
$1.4641\,0016\,947_{10}+4$	$4.5697\,6000\,342_{10}+5$		
$2.0736\,0014\,498_{10}+4$	$5.3144\,1000\,319_{10}+5$		
$2.8561\,0012\,540_{10}+4$	$6.1465\,6000\,296_{10}+5$		
$3.8416\,0010\,949_{10}+4$	$7.0728\,1000\,278_{10}+5$		
$5.0625\,0009\,644_{10}+4$	$8.1000\,0008\,188_{10}+5$		
$eps2 = 1.0319_{10}-5$		$eps2 = 1.0314_{10}-5$	$eps2 = 1.0314_{10}-5$

With a value of *eps1* of 10^{-12} all eigenvalues were given correctly to within five units in the last significant figure; this represents a far smaller error than *eps2* in the smaller eigenvalues.

The final matrix is defined by

$$c_i = 110 - 10i \quad (i = 1, \ldots, 11), \qquad c_i = 10i - 110 \quad (i = 12, \ldots, 21),$$
$$b_i = 1 \quad (i = 2, \ldots, 21).$$

This matrix has a number of extremely close (but not coincident) eigenvalues. It was solved on KDF 9 using $eps1 = 10^{-7}$ and the results are given in Table 2.

Table 2

$-1.9709\,2929\,389_{10}-1$	$5.9999\,9999\,\underline{5}39_{10}+1$
$9.9004\,9425\,559_{10}+0$	$5.9999\,9999\,\underline{5}39_{10}+1$
$1.0096\,5954\,\underline{7}03_{10}+1$	$7.0000\,0000\,\underline{4}70_{10}+1$
$1.9999\,5066\,\underline{1}65_{10}+1$	$7.0000\,0000\,\underline{4}70_{10}+1$
$2.0000\,4966\,\underline{7}11_{10}+1$	$8.0000\,0008\,\underline{1}16_{10}+1$
$2.9999\,9991\,\underline{9}49_{10}+1$	$8.0000\,0008\,\underline{1}16_{10}+1$
$3.0000\,0008\,\underline{2}56_{10}+1$	$9.0000\,4933\,\underline{9}00_{10}+1$
$3.9999\,9999\,\underline{5}95_{10}+1$	$9.0000\,4933\,\underline{9}00_{10}+1$
$3.9999\,9999\,\underline{5}95_{10}+1$	$1.0009\,9505\,7\underline{5}1_{10}+2$
$4.9999\,9999\,\underline{5}67_{10}+1$	$1.0009\,9505\,7\underline{5}1_{10}+2$
$4.9999\,9999\,\underline{5}67_{10}+1$	$eps2 = 1.0128_{10}-7$

The error in each computed values is a few units in the underlined figure and is in each case less than $\frac{1}{2}10^{-7}$ in absolute magnitude. The total number of iteration was 345 an average of 16.4 per eigenvalue. It should be appreciated that by using a smaller value of *eps1* the results can be obtained correct to working accuracy.

Acknowledgements. The authors wish to thank Professor Dr. H. RUTISHAUSER for important suggestions, and Dipl.-Math. L. KRAUSS for help in writing and testing a preliminary version of the procedure made at Darmstadt. The contribution by R. S. MARTIN and J. H. WILKINSON has been carried out at the National Physical Laboratory.

References

[1] BOWDLER, H. J., C. REINSCH, and J. H. WILKINSON: The QR algorithm for symmetric tridiagonal matrices (to be published in this series).

[2] GIVENS, J. W.: Numerical computation of the characteristic values of a real symmetric matrix. Oak Ridge National Laboratory, ORNL-1574 (1954).

[3] RUTISHAUSER, H.: Stabile Sonderfälle des Quotienten-Differenzen-Algorithmus. Numerische Mathematik **5**, 95—11 (1963).

[4] WILKINSON, J. H.: Error analysis of floating-point computation. Numerische Mathematik **2**, 319—340 (1960).

[5] — Calculation of the eigenvalue of a symmetric tridiagonal matrix by the method of bisection. Numerische Mathematik **4**, 362—367 (1962).

[6] — The algebraic eigenvalue problem. London: Oxford University Press 1965.

Rechenzentrum
Technische Hochschule
Darmstadt

National Physical Laboratory
Teddington, Middlesex
Great Britain