

Lubomír Skála

Algorithms. 42. BISEC. The eigenvalues of the symmetric eigenproblem $\mathbf{A}x = \lambda\mathbf{B}x$ and related eigenproblems

Aplikace matematiky, Vol. 22 (1977), No. 6, 463--469

Persistent URL: <http://dml.cz/dmlcz/103721>

Terms of use:

© Institute of Mathematics AS CR, 1977

Institute of Mathematics of the Academy of Sciences of the Czech Republic provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This paper has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* <http://project.dml.cz>

ALGORITMY

42. BISEC

THE EIGENVALUES OF THE SYMMETRIC EIGENPROBLEM $\mathbf{A}x = \lambda \mathbf{B}x$
AND RELATED EIGENPROBLEMS

Dr. LUBOMÍR SKÁLA, Matematicko-fyzikální fakulta KU,
Ke Karlovu 3, 121 16 Praha 2.

In some cases, the eigenvalues of a generalized eigenvalue problem

$$(1) \quad \mathbf{A}x = \lambda \mathbf{B}x,$$

where $\mathbf{A} = \{\mathbf{A}_{ij}\}_{i,j=1,\dots,n}$ and $\mathbf{B} = \{\mathbf{B}_{ij}\}_{i,j=1,\dots,n}$ are real symmetric matrices of order n , \mathbf{B} positive definite and of related eigenproblems

$$(2) \quad \mathbf{A}^{(k)}x = \lambda \mathbf{B}^{(k)}x, \quad k = 1, \dots, n - 1,$$

where

$$(3) \quad \mathbf{A}^{(k)} = \{\mathbf{A}_{ij}\}_{i,j=1,\dots,k} \quad \text{and} \quad \mathbf{B}^{(k)} = \{\mathbf{B}_{ij}\}_{i,j=1,\dots,k}$$

are submatrices of \mathbf{A} and \mathbf{B} , are required. An efficient method for calculating eigenvalues of all eigenproblems (1) and (2) is given here.

For a given real number μ , we decompose the matrix $\mathbf{A} - \mu\mathbf{B}$ as

$$(4) \quad \mathbf{A} - \mu\mathbf{B} = \mathbf{L} \cdot \mathbf{U},$$

where $\mathbf{L}(\mathbf{U})$ is the lower (upper) triangle matrix and $\mathbf{L}_{ii} = 1$. Then, the number of eigenvalues of the eigenproblem $\mathbf{A}^{(k)}x = \lambda \mathbf{B}^{(k)}x$ less than μ is equal to (for $\mathbf{B} = 1$ see e.g. [1]; for $\mathbf{B} \neq 1$, the generalization is straightforward)

$$(5) \quad I_k(\mu) = \sum_{i=1}^k \Theta(-\mathbf{U}_{ii}).$$

Here, $\Theta(x) = 1$ for $x > 0$, $\Theta(x) = 0$ otherwise. Making use of (5), we calculate the eigenvalues of (1) and (2) by the method of bisection (the procedure *bisec*). The

quantities $I_k(\mu)$ are calculated by the procedure *ik* (see also [2]). Some information obtained when determining one eigenvalue of (1) or (2) (for a given k) is, in general, of significance in the determination of other eigenvalues of the same eigenproblem (see [3]). In *bisec*, full advantage is taken of all relevant information and this results in a very substantial saving of time in case of a number of close or coincident eigenvalues. According to (5), the calculation of $I_k(\mu)$ gives the values of $I_1(\mu), \dots, I_{k-1}(\mu)$ as an additional result. The full use of this information further reduces the computational time.

The procedure *bisec* may be used to calculate the eigenvalues of all eigenproblems (1) and (2) in a given interval (e_{\min}, e_{\max}) . The matrices **A** and **B** are assumed to be band matrices ($a_{ij} = b_{ij} = 0$ for $|i - j| \geq m, m \leq n$). For the sake of efficiency, just the (i, j) elements ($0 \leq i - j < m$) of these matrices are stored, column by column, in one-dimensional arrays *a* and *b*. The number of arithmetic operations in one bisection step is about $1.5nm^2$. Therefore, the procedure *bisec* should not be used for band matrices with high half-bandwidth m . In general, it works most efficiently if the order n is very high, since the computational time is then roughly given by the time needed to calculate the eigenvalues of a few eigenproblems of the greatest order. The procedure *bisec* is in such cases much more efficient than other methods which calculate the eigenvalues of each eigenproblem separately. The accuracy of results is not influenced by close or coincident eigenvalues.

For large n , it is usually impossible to store all upper bounds e_{kl} (the upper bound to the l -th eigenvalue of the eigenproblem of the k -th order) or, respectively, lower bounds d_{kl} in the core. Hence, we store these two arrays (which are for the sake of simplicity assumed to be square arrays) on a disc row by row. The procedure *Read* ($k, d1, e1$) reads the k -th row of the arrays *e* and *d* from a disc and stores them in the arrays *e1* and *d1* in the core. Similarly, the procedure *Write* ($k, d1, e1$) stores the arrays *e1* and *d1* from the core in the k -th row of the arrays *e* and *d* on a disc, respectively.

procedure *ik*(*a, b, mi, m, n, eps, rel, q*);

value *mi, m, n, eps, rel*;

real *mi, eps, rel*;

integer *m, n*;

real array *a, b*;

integer array *q*;

comment Input to procedure *ik*

a, b $[(n - m + 1) \times m + (m - 1) \times m//2] \times 1$ arrays giving the (i, j) elements ($0 \leq i - j < m$) of the matrices **A** and **B** stored column by column.

mi the number of eigenvalues less than *mi* will be found.

m bandwidth of **A** and **B** is $2m - 1$.

n order of **A** and **B**.

eps the smallest positive real number representable on the computer.
rel the smallest positive real number for which $1 + rel > 1$ on the computer.

Output of procedure *ik*

q $n \times 1$ array. $q[k]$ gives the number of eigenvalues less than m_i for the eigenproblem of the k -th order;

```

begin
  real c, x, ymax;
  integer i, i1, i2, j, k, l1, l2, n2;
  real array y[1 : m], w[1 : (n - m + 1) × m + (m - 1) × m//2];
  i2 := (n - m + 1) × m + (m - 1) × m//2;
  for i := 1 step 1 until i2 do w[i] := a[i] - mi × b[i];
  x := w[1];
  if x < eps then q[1] := 1 else q[1] := 0;
  i1 := 1;
  for i := 2 step 1 until n do
  begin n2 := n - i + 2;
    l1 := n2 - m;
    if l1 < 0 then l1 := 0;
    if m < n2 then n2 := m;
    ymax := 0;
    for j := 2 step 1 until n2 do
    begin i1 := i1 + 1;
      y[j] := c := w[i1];
      if abs(c) > ymax then ymax := abs(c)
    end j;
    if ymax ≥ eps then
    begin i2 := i1;
      for j := 2 step 1 until n2 do
      begin if abs(x) < eps then c := y[j]/ymax/rel
        else c := -y[j]/x;
        for k := j step 1 until n2 do
        begin i2 := i2 + 1;
          w[i2] := w[i2] + c × y[k]
        end k;
        l2 := j - 1;
        if l2 < l1 then i2 := i2 + l2 else i2 := i2 + l1
        end j
      end;
    i1 := i1 + 1;
    x := w[i1];
    if x < eps then q[i] := q[i - 1] + 1 else q[i] := q[i - 1];
  end

```

```

end i
end ik;
procedure bisec (a, b, m, n, epsres, emin, emax, eps, rel) result: (m1, m2) exit: (fail);
value m, n, epsres, emin, emax, eps, rel;
real epsres, emin, emax, eps, rel;
integer m, n;
real array a, b;
integer array m1, m2;
label fail;
comment Input to procedure bisec
    a, b       $[(n - m + 1) \times m + (m - 1) \times m/2] \times 1$  arrays giving the
              (i, j) elements ( $0 \leq i - j < m$ ) of the matrices A and B
              stored column by column.
    m        bandwidth of A and B is  $2m - 1$ .
    n        order of A and B.
    epsres    the relative (for an eigenvalue  $> 1$ ) or the absolute (for an
              eigenvalue  $< 1$ ) error in any eigenvalue.
    emin, emax all eigenvalues in (emin, emax) will be computed.
    eps      the smallest positive real number representable on the com-
              puter.
    rel      the smallest positive real number for which  $1 + rel > 1$  on
              the computer.

    Output of procedure bisec
    m1, m2    $n \times 1$  arrays. For an eigenproblem of order k, the eigenvalues
              with sequential numbers  $m1[k], \dots, m2[k]$  lie in (emin, emax).

    e         $n \times n$  array stored on a disc gives the computed eigenvalues.
              The l-th eigenvalue of the k-th eigenproblem is stored as  $e_{kl}$ .
              For each eigenproblem, the eigenvalues are arranged in
              ascending order.

    fail     the exit used if B is not positive definite;
begin real g, h, mi; integer i, i1, i2, j, k, l, qk;
real array d1, e1, d2, e2[1 : n]; integer array q[1 : n];
ik(b, b, 0, m, n, eps, rel, q);
if  $q[n] > 0$  then go to fail;
comment The calculation of sequential numbers of the eigenvalues lying in (emin, emax);
ik(a, b, emin, m, n, eps, rel, m1);
ik(a, b, emax, m, n, eps, rel, m2);
for  $i := 1$  step 1 until  $n$  do
begin  $m1[i] := m1[i] + 1$ ;

```

```

    for j := m1[i] step 1 until m2[i] do
    begin d1[j] := emin;
         e1[j] := emax
    end j;
    Write(i, d1, e1)
end i;
for k := n step -1 until 1 do
begin
comment The calculation of the eigenvalues for the eigenproblem of the k-th order;
Read(k, d1, e1);
if m > k then m := k;
h := emax;
comment Loop for the l-th eigenvalue;
for l := m2[k] step -1 until m1[k] do
begin g := emin;
    for i := l step -1 until m1[k] do
    begin if g < d1[i] then
        begin g := d1[i];
             go to cont
        end
    end i;
cont: if h > e1[l] then h := e1[l];
comment The method of bisection;
for mi := (g + h)/2 while h - g > 2 × (epsres × abs(mi) + epsres) do
begin ik(a, b, mi, m, k, eps, rel, q);
    qk := q[k];
    if qk < l then
    begin if qk < m1[k] then d1[m1[k]] := g := mi
        else
        begin g := d1[qk + 1] := mi;
            if e1[qk] > mi then e1[qk] := mi
        end
    end
    else h := mi;
comment The calculation of new lower and upper bounds;
for i := k - 1 step -1 until 1 do
begin
Read(i, d2, e2);
for j := m1[i] step 1 until m2[i] do
if q[i] < j then
begin if d2[j] < mi then d2[j] := mi end
else

```

```

        if e2[j] > mi then e2[j] := mi;
        Write(i, d2, e2)
    end i
end mi;
e1[l] := mi
end l;
comment New storage of a and b;
i1 := i2 := 0;
for i := 1 step 1 until k do
begin l := k - i + 1;
    if m < l then l := m;
    for j := i step 1 until i + l - 1 do
begin i1 := i1 + 1;
    if j < k then
begin i2 := i2 + 1;
        a[i2] := a[i1];
        b[i2] := b[i1]
    end
    end j
end i;
Write(k, d1, e1)
end k
end bisec;

```

Example.

For the data $a = (10, 2, 3, 12, 1, 2, 11, 1, 9)$,
 $b = (12, 1, -1, 14, 1, -1, 16, -1, 12)$, $m = 3$, $n = 4$, $epsres = 10^{-15}$,
 $emin = -10$, $emax = 10$, $eps = 10^{-75}$, $rel = 10^{-15}$ corresponding to the matrices

$$\mathbf{A} = \begin{pmatrix} 10 & 2 & 3 & 0 \\ 2 & 12 & 1 & 2 \\ 3 & 1 & 11 & 1 \\ 0 & 2 & 1 & 9 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 12 & 1 & -1 & 0 \\ 1 & 14 & 1 & -1 \\ -1 & 1 & 16 & -1 \\ 0 & -1 & -1 & 12 \end{pmatrix}$$

we obtained the following results (the computer ICL 4-72)

$$\begin{aligned}
 m1[1] &= m1[2] = m1[3] = m1[4] = 1, \\
 m2[1] &= 1, \quad m2[2] = 2, \quad m2[3] = 3, \quad m2[4] = 4, \\
 e[1, 1] &= 8.3333 \ 3333 \ 3333 \ 337_{10} - 1, \\
 e[2, 1] &= 7.4790 \ 6174 \ 4278 \ 959_{10} - 1, \\
 e[2, 2] &= 9.2874 \ 0532 \ 1589 \ 304_{10} - 1, \\
 e[3, 1] &= 4.9264 \ 3004 \ 8161 \ 612_{10} - 1, \\
 e[3, 2] &= 8.3439 \ 0032 \ 4405 \ 518_{10} - 1,
 \end{aligned}$$

$$\begin{aligned}
e[3, 3] &= 1.0765\ 2218\ 2107\ 887, \\
e[4, 1] &= 4.4739\ 1135\ 7782\ 800_{10} - 1, \\
e[4, 2] &= 6.5396\ 6400\ 2667\ 978_{10} - 1, \\
e[4, 3] &= 9.4074\ 1722\ 5080\ 661_{10} - 1, \\
e[4, 4] &= 1.1602\ 1950\ 8168\ 733.
\end{aligned}$$

References

- [1] *P. Dean* The Vibrational Properties of Disordered Systems: Numerical Studies. *Rev. Mod. Phys.* 44 (1972), 127–168.
- [2] *L. Skála* NEGEIG. Distribution of the Eigenvalues of Eigenproblem $Ax = \lambda Bx$. *Aplikace matematiky* 20 (1975), 227–231.
- [3] *W. Barth, R. S. Martin, J. H. Wilkinson* Calculation of the Eigenvalues of a Symetric Tri-diagonal Matrix by the Method of Bisections. *Numer. Math.* 9 (1967), 386–393.