Document reference:


MSC 65F15. [DMLCZ 103721](http://dml.cz/dmlcz/103721).


Terms of use:

© Institute of Mathematics, Academy of Sciences of the Czech Republic, 2008

Institute of Mathematics, 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.

[Czech Digital Mathematics Library](http://project.dml.cz)

This paper has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library*
42. BISEC

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

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

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

\[(2) \quad \mathbf{A}^{(k)} \mathbf{x} = \lambda \mathbf{B}^{(k)} \mathbf{x}, \quad k = 1, \ldots, n - 1,\]

where

\[(3) \quad \mathbf{A}^{(k)} = \{A_{ij}\}_{i,j=1,...,k} \quad \text{and} \quad \mathbf{B}^{(k)} = \{B_{ij}\}_{i,j=1,...,k}\]

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 $\mu$, we decompose the matrix $\mathbf{A} - \mu \mathbf{B}$ as

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

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

\[(5) \quad I_k(\mu) = \sum_{i=1}^{k} \Theta(-U_{ii}).\]

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 $I_k(h)$ 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 $I_k(h)$ gives the values of $I_1(h), \ldots, I_{k-1}(h)$ 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 $ik$**

$a, b \left[ (n - m + 1) \times m + (m - 1) \times m/2 \right] \times 1$ 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.

$m$ bandwith 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 $mi$ for the eigenproblem of the $k$-th order;

begin
real $c$, $x$, $y_{\text{max}}$;
integer $i$, $i_1$, $i_2$, $j$, $k$, $l_1$, $l_2$, $n_2$;
real array $y[1 : m]$, $w[1 : (n - m + 1) \times m + (m - 1) \times m/2]$;
i_2 := (n - m + 1) \times m + (m - 1) \times m/2;
for $i := 1$ step 1 until $i_2$ do $w[i] := a[i] - mi \times b[i]$;
x := $w[1]$;
if $x < eps$ then $q[1] := 1$ else $q[1] := 0$;
i_1 := 1;
for $i := 2$ step 1 until $n$ do
begin
i_2 := i_1 + 2;
l_1 := n_2 - m;
if $l_1 < 0$ then $l_1 := 0$;
if $m < n_2$ then $n_2 := m$;
y_{\text{max}} := 0;
for $j := 2$ step 1 until $n_2$ do
begin
i_1 := i_1 + 1;
y[j] := $c := w[i_1]$;
if $\text{abs}(c) > y_{\text{max}}$ then $y_{\text{max}} := \text{abs}(c)$
endj;
if $y_{\text{max}} \geq eps$ then
begin
i_2 := i_1;
for $j := 2$ step 1 until $n_2$ do
begin
if $\text{abs}(x) < eps$ then $c := y[j]/y_{\text{max}}/rel$
else $c := -y[j]/x$;
for $k := j$ step 1 until $n_2$ do
begin
i_2 := i_2 + 1;
w[i_2] := $w[i_2] + c \times y[k]$;
endk;
i_2 := j - 1;
if $l_2 < l_1$ then $i_2 := i_2 + l_2$ else $i_2 := i_2 + l_1$
endj;
i_1 := i_1 + 1;
x := $w[i_1]$;
if $x < eps$ then $q[i] := q[i - 1] + 1$ else $q[i] := q[i - 1]$;
end;
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;

Input to procedure bisec

\[(a,b, m, n, \text{epsres}, \text{emin}, \text{emax}, \text{eps}, \text{rel})\]

\(\text{m} \times \text{n} - \text{m} + 1\) \times \text{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 \(B\).

\(\text{epsres}\) the relative (for an eigenvalue >1) or the absolute (for an
eigenvalue <1) error in any eigenvalue.

\(\text{emin}, \text{emax}\) all eigenvalues in \((\text{emin}, \text{emax})\) will be computed.

\(\text{eps}\) the smallest positive real number representable on the com­
puter.

\(\text{rel}\) the smallest positive real number for which \(1 + \text{rel} > 1\) on
the computer.

Output of procedure bisec

\(m1, m2\) \(n \times 1\) arrays. For an eigenproblem of order \(k\), the eigenvalues
with sequentional numbers \(m1[k], ..., m2[k]\) lie in \((\text{emin}, \text{emax})\).

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

\(\text{fail}\) the exit used if \(B\) is not positive definite;

\begin{verbatim}
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 sequentional 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;
end i
end ik;
end bisec;
\end{verbatim}
for $j := ml[i]$ 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
$g := d1[i]$;
end
end $i$;
cont: if $h > e1[l]$ then $h := e1[l]$;
comment The method of bisection;
for $mi := (g + h)/2$ while $h - g > 2 \times (epsres \times abs(mi) + epsres)$ do
begin
$qk := q[k]$;
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
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 \( e_2[j] > m_i \) then \( e_2[j] := m_i \);
\[
\text{Write}(i, d_2, e_2)
\]
end \( i \)
end \( m_i \);
\[
e_1[l] := m_i
\]
end \( l \);

comment New storage of \( a \) and \( b \);
\[
i_1 := i_2 := 0;
\]
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 \( i_1 := i_1 + 1 \);
if \( j < k \) then
begin \( i_2 := i_2 + 1 \);
\[
a[i_2] := a[i_1];
b[i_2] := b[i_1]
\]
end
end
end \( i \);
\[
\text{Write}(k, d_1, e_1)
\]
end \( k \)
end bisec;

Example.

For the data \( a = (10, 2, 3, 12, 1, 2, 11, 1, 9) \),
\( b = (12, 1, -1, 14, 1, -1, 16, -1, 12) \), \( m = 3 \), \( n = 4 \), \( \epsilon_{\text{res}} = 10^{-15} \),
\( \epsilon_{\text{min}} = -10 \), \( \epsilon_{\text{max}} = 10 \), \( \epsilon = 10^{-75} \), \( \epsilon_{\text{rel}} = 10^{-15} \) corresponding to the matrices
\[
A = \begin{pmatrix}
10 & 2 & 3 & 0 \\
2 & 12 & 1 & 2 \\
3 & 1 & 11 & 1 \\
0 & 2 & 1 & 9
\end{pmatrix}, \quad B = \begin{pmatrix}
12 & 1 & -1 & 0 \\
1 & 14 & 1 & -1 \\
-1 & 1 & 16 & -1 \\
0 & -1 & -1 & 12
\end{pmatrix}
\]

we obtained the following results (the computer ICL 4-72)
\[
m_2[1] = 1, \quad m_2[2] = 2, \quad m_2[3] = 3, \quad m_2[4] = 4,
e[1, 1] = 8 \cdot 333333333333337_{10}, -1,
e[2, 1] = 7.4790 \cdot 61744278959_{10}, -1,
e[2, 2] = 9 \cdot 287405321589304_{10}, -1,
e[3, 1] = 4.9264 \cdot 30048161612_{10}, -1,
e[3, 2] = 8.343900324405518_{10}, -1,
\]
\[ e_{[3, 3]} = 1.0765 \quad 2218 \quad 2107 \quad 887, \]
\[ e_{[4, 1]} = 4.4739 \quad 1135 \quad 7782 \quad 800_{10} - 1, \]
\[ e_{[4, 2]} = 6.5396 \quad 6400 \quad 2667 \quad 978_{10} - 1, \]
\[ e_{[4, 3]} = 9.4074 \quad 1722 \quad 5080 \quad 661_{10} - 1, \]
\[ e_{[4, 4]} = 1.1602 \quad 1950 \quad 8168 \quad 733. \]

References