NAG Toolbox for MATLAB

f08sb

1 Purpose

f08sb computes selected eigenvalues and, optionally, eigenvectors of a real generalized symmetric-definite eigenproblem, of the form

$$Az = \lambda Bz, \qquad ABz = \lambda z \qquad \text{or} \qquad BAz = \lambda z,$$

where A and B are symmetric and B is also positive-definite. Eigenvalues and eigenvectors can be selected by specifying either a range of values or a range of indices for the desired eigenvalues.

2 Syntax

```
[a, b, m, w, z, jfail, info] = f08sb(itype, jobz, range, uplo, a, b, vl,
vu, il, iu, abstol, 'n', n)
```

3 Description

f08sb first performs a Cholesky factorization of the matrix B as $B = U^{\mathrm{T}}U$, when **uplo** = 'U' or $B = LL^{\mathrm{T}}$, when **uplo** = 'L'. The generalized problem is then reduced to a standard symmetric eigenvalue problem

$$Cx = \lambda x,$$

which is solved for the desired eigenvalues and eigenvectors. The eigenvectors of C are then backtransformed to give the eigenvectors of the original problem.

For the problem $Az = \lambda Bz$ and $ABz = \lambda z$, the eigenvectors are normalized so that

$$z^{\mathrm{T}}Bz = I.$$

For the problem $BAz = \lambda z$ we correspondingly have

$$z^{\mathrm{T}}B^{-1}z = I.$$

4 References

Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J J, Du Croz J J, Greenbaum A, Hammarling S, McKenney A and Sorensen D 1999 *LAPACK Users' Guide* (3rd Edition) SIAM, Philadelphia URL: http://www.netlib.org/lapack/lug

Demmel J W and Kahan W 1990 Accurate singular values of bidiagonal matrices *SIAM J. Sci. Statist. Comput.* **11** 873–912

Golub G H and Van Loan C F 1996 *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

5 Parameters

5.1 Compulsory Input Parameters

1: **itype – int32 scalar**

Specifies the problem type to be solved.

itype $= 1$

$Az = \lambda Bz.$

itype $= 2$

$ABz = \lambda z.$

itype $= 3$

$BAz = \lambda z.$

2: **jobz – string**

If **jobz** = 'N', compute eigenvalues only.

If **jobz** = 'V', compute eigenvalues and eigenvectors.

Constraint: **jobz** = 'N' or 'V'.

3: **range – string**

If **range** = 'A', all eigenvalues will be found.

If **range** = 'V', all eigenvalues in the half-open interval (**vl**, **vu**] will be found.

If **range** = 'I', the **il**th to **iu**th eigenvalues will be found.

Constraint: **range** = 'A', 'V' or 'I'.

4: **uplo – string**

If **uplo** = 'U', the upper triangles of A and B are stored.

if **uplo** = 'L', the lower triangles of A and B are stored.

Constraint: **uplo** = 'U' or 'L'.

5: **a(lda,*) – double array**

The first dimension of the array **a** must be at least $\max(1,\mathbf{n})$

The second dimension of the array must be at least $\max(1,\mathbf{n})$

The n by n symmetric matrix A.

If **uplo** = 'U', the upper triangular part of A must be stored and the elements of the array below the diagonal are not referenced.

If **uplo** = 'L', the lower triangular part of A must be stored and the elements of the array above the diagonal are not referenced.

6: **b(ldb,*) – double array**

The first dimension of the array **b** must be at least $\max(1,\mathbf{n})$

The second dimension of the array must be at least $\max(1,\mathbf{n})$

The symmetric matrix B:

if **uplo** = 'U', the leading n by n upper triangular part of **b** contains the upper triangular part of the matrix B;
if **uplo** = 'L', the leading n by n lower triangular part of **b** contains the lower triangular part of the matrix B.

7: **vl – double scalar**
8: **vu – double scalar**

If **range** = 'V', the lower and upper bounds of the interval to be searched for eigenvalues.

If **range** $=$ 'A' or 'I', **vl** and **vu** are not referenced.

Constraint: if **range** $=$ 'V', **vl** $<$ **vu**.

9: **il – int32 scalar**

10: **iu – int32 scalar**

If **range** $=$ 'I', the indices (in ascending order) of the smallest and largest eigenvalues to be returned.

If **range** $=$ 'A' or 'V', **il** and **iu** are not referenced.

Constraints:

if **n** $= 0$, **il** $= 1$ and **iu** $= 0$;
if **n** > 0, $1 \leq$ **il** $\leq$ **iu** $\leq$ **n**.

11: **abstol – double scalar**

The absolute error tolerance for the eigenvalues. An approximate eigenvalue is accepted as converged when it is determined to lie in an interval $[a, b]$ of width less than or equal to

$$\mathbf{abstol} + \epsilon \max(|a|, |b|),$$

where ϵ is the ***machine precision***. If **abstol** is less than or equal to zero, then $\epsilon \|T\|_1$ will be used in its place, where T is the tridiagonal matrix obtained by reducing C to tridiagonal form. Eigenvalues will be computed most accurately when **abstol** is set to twice the underflow threshold $2 \times$ x02am( ), not zero. If this function returns with **info** > 0, indicating that some eigenvectors did not converge, try setting **abstol** to $2 \times$ x02am( ). See Demmel and Kahan 1990.

5.2 Optional Input Parameters

1: **n – int32 scalar**

Default: The second dimension of the array **a** The second dimension of the array **b**.

n, the order of the matrix pencil (A, B).

Constraint: **n** ≥ 0.

5.3 Input Parameters Omitted from the MATLAB Interface

lda, ldb, ldz, work, lwork, iwork

5.4 Output Parameters

1: **a(lda,$*$) – double array**

The first dimension of the array **a** must be at least $\max(1, \mathbf{n})$

The second dimension of the array must be at least $\max(1, \mathbf{n})$

The lower triangle (if **uplo** $=$ 'L') or the upper triangle (if **uplo** $=$ 'U') of **a**, including the diagonal, is destroyed.

2: **b(ldb,$*$) – double array**

The first dimension of the array **b** must be at least $\max(1, \mathbf{n})$

The second dimension of the array must be at least $\max(1, \mathbf{n})$

If **info** $\leq$ **n**, the part of **b** containing the matrix contains the triangular factor U or L from the Cholesky factorization $\mathbf{b} = U^{\mathrm{T}}U$ or $\mathbf{b} = LL^{\mathrm{T}}$.

3: **m – int32 scalar**

The total number of eigenvalues found.

If **range** $=$ 'A', **m** $=$ **n**.