NAG Toolbox for MATLAB

f08tc

1 Purpose

f08tc computes all the eigenvalues and, optionally, the 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, stored in packed format, and B is also positive-definite. If eigenvectors are desired, it uses a divide-and-conquer algorithm.

2 Syntax

```
[ap, bp, w, z, info] = f08tc(itype, jobz, uplo, n, ap, bp)
```

3 Description

f08tc 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 eigenvalues and, optionally, the eigenvectors; the eigenvectors are then backtransformed to give the eigenvectors of the original problem.

For the problem $Az = \lambda Bz$, the eigenvectors are normalized so that the matrix of eigenvectors, z, satisfies

$$Z^{\mathrm{T}}AZ = \Lambda \qquad \text{and} \qquad Z^{\mathrm{T}}BZ = I,$$

where Λ is the diagonal matrix whose diagonal elements are the eigenvalues. For the problem $ABz = \lambda z$ we correspondingly have

$$Z^{-1}AZ^{-\mathrm{T}} = \Lambda \qquad \text{and} \qquad Z^{\mathrm{T}}BZ = I,$$

and for $BAz = \lambda z$ we have

$$Z^{\mathrm{T}}AZ = \Lambda \qquad \text{and} \qquad Z^{\mathrm{T}}B^{-1}Z = I.$$

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

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: **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'.

4: **n – int32 scalar**

n, the order of the matrices A and B.

Constraint: $\mathbf{n} \geq 0$.

5: **ap**$(*)$ **– double array**

Note: the dimension of the array **ap** must be at least $\max(1, \mathbf{n} \times (\mathbf{n} + 1)/2)$.

The n by n symmetric matrix A, packed by columns.

More precisely,

if **uplo** = 'U', the upper triangle of A must be stored with element A_{ij} in $\mathbf{ap}(i + j(j - 1)/2)$ for $i \leq j$;
if **uplo** = 'L', the lower triangle of A must be stored with element A_{ij} in $\mathbf{ap}(i + (2n - j)(j - 1)/2)$ for $i \geq j$.

6: **bp**$(*)$ **– double array**

Note: the dimension of the array **bp** must be at least $\max(1, \mathbf{n} \times (\mathbf{n} + 1)/2)$.

The upper or lower triangle of the symmetric matrix B, packed by columns.

More precisely,

if **uplo** = 'U', the upper triangle of B must be stored with element B_{ij} in $\mathbf{bp}(i + j(j - 1)/2)$ for $i \leq j$;
if **uplo** = 'L', the lower triangle of B must be stored with element B_{ij} in $\mathbf{bp}(i + (2n - j)(j - 1)/2)$ for $i \geq j$.

5.2 Optional Input Parameters

None.

5.3 Input Parameters Omitted from the MATLAB Interface

ldz, work, lwork, iwork, liwork

5.4 Output Parameters

1: **ap**($*$) – **double array**

Note: the dimension of the array **ap** must be at least $\max(1, \mathbf{n} \times (\mathbf{n}+1)/2)$.

The contents of **ap** are destroyed.

2: **bp**($*$) – **double array**

Note: the dimension of the array **bp** must be at least $\max(1, \mathbf{n} \times (\mathbf{n}+1)/2)$.

The triangular factor U or L from the Cholesky factorization $B = U^{\mathrm{T}}U$ or $B = LL^{\mathrm{T}}$, in the same storage format as B.

3: **w**($*$) – **double array**

Note: the dimension of the array **w** must be at least $\max(1, \mathbf{n})$.

If **info** $= 0$, the eigenvalues in ascending order.

4: **z**(**ldz**,$*$) – **double array**

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

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

If **jobz** = 'V', then if **info** $= 0$, **z** contains the matrix Z of eigenvectors. The eigenvectors are normalized as follows:

if **itype** $= 1$ or 2, $Z^{\mathrm{T}}BZ = I$;
if **itype** $= 3$, $Z^{\mathrm{T}}B^{-1}Z = I$.

If **jobz** = 'N', **z** is not referenced.

5: **info** – **int32 scalar**

info $= 0$ unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

info $= -i$

If **info** $= -i$, parameter i had an illegal value on entry. The parameters are numbered as follows:

1: **itype**, 2: **jobz**, 3: **uplo**, 4: **n**, 5: **ap**, 6: **bp**, 7: **w**, 8: **z**, 9: **ldz**, 10: **work**, 11: **lwork**, 12: **iwork**, 13: **liwork**, 14: **info**.

It is possible that **info** refers to a parameter that is omitted from the MATLAB interface. This usually indicates that an error in one of the other input parameters has caused an incorrect value to be inferred.

info > 0

f07gd or f08gc returned an error code:

$\leq$ **n** if **info** $= i$, f08gc failed to converge; i off-diagonal elements of an intermediate tridiagonal form did not converge to zero;

$>$ **n** if **info** $= \mathbf{n} + i$, for $1 \leq i \leq \mathbf{n}$, then the leading minor of order i of B is not positive-definite. The factorization of B could not be completed and no eigenvalues or eigenvectors were computed.

7 Accuracy

If B is ill-conditioned with respect to inversion, then the error bounds for the computed eigenvalues and vectors may be large, although when the diagonal elements of B differ widely in magnitude the eigenvalues and eigenvectors may be less sensitive than the condition of B would suggest. See Section 4.10 of Anderson *et al.* 1999 for details of the error bounds.

The example program below illustrates the computation of approximate error bounds.

8 Further Comments

The total number of floating-point operations is proportional to n^3.

The complex analogue of this function is f08tq.

9 Example

```
itype = int32(2);
jobz = 'No vectors';
uplo = 'U';
n = int32(4);
ap = [0.24;
     0.39;
     -0.11;
     0.42;
     0.79;
     -0.25;
     -0.16;
     0.63;
     0.48;
     -0.03];
bp = [4.16;
     -3.12;
     5.03;
     0.56;
     -0.83;
     0.76;
     -0.1;
     1.09;
     0.34;
     1.18];
[apOut, bpOut, w, z, info] = f08tc(itype, jobz, uplo, n, ap, bp)
```

```
apOut =
    0.1373
    0.9473
   -0.2325
   -0.7098
   -2.5252
   -1.2164
   -0.6202
    0.4797
   -0.9277
   -0.0114
bpOut =
    2.0396
   -1.5297
    1.6401
    0.2746
   -0.2500
    0.7887
   -0.0490
    0.6189
    0.6443
    0.6161
```

```
w =
   -3.5411
   -0.3347
    0.2983
    2.2544
z =
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
info =
           0
```