F08TEF (SSPGST/DSPGST) – NAG Fortran Library Routine Document

Note. Before using this routine, please read the Users' Note for your implementation to check the interpretation of bold italicised terms and other implementation-dependent details.

1 Purpose

F08TEF (SSPGST/DSPGST) reduces a real symmetric-definite generalized eigenproblem $Az = \lambda Bz$, $ABz = \lambda z$ or $BAz = \lambda z$ to the standard form $Cy = \lambda y$, where A is a real symmetric matrix and B has been factorized by F07GDF (SPPTRF/DPPTRF), using packed storage.

2 Specification

```
SUBROUTINE F08TEF(ITYPE, UPLO, N, AP, BP, INFO)
ENTRY      sspgst(ITYPE, UPLO, N, AP, BP, INFO)
INTEGER           ITYPE, N, INFO
real              AP(*), BP(*)
CHARACTER*1       UPLO
```

The ENTRY statement enables the routine to be called by its LAPACK name.

3 Description

To reduce the real symmetric-definite generalized eigenproblem $Az = \lambda Bz$, $ABz = \lambda z$ or $BAz = \lambda z$ to the standard form $Cy = \lambda y$ using packed storage, this routine must be preceded by a call to F07GDF (SPPTRF/DPPTRF) which computes the Cholesky factorization of B; B must be positive-definite.

The different problem types are specified by the parameter ITYPE, as indicated in the table below. The table shows how C is computed by the routine, and also how the eigenvectors z of the original problem can be recovered from the eigenvectors of the standard form.

ITYPE	Problem	UPLO	B	C	z
1	$Az = \lambda Bz$	'U' 'L'	U^TU LL^T	$U^{-T}AU^{-1}$ $L^{-1}AL^{-T}$	$U^{-1}y$ $L^{-T}y$
2	$ABz = \lambda z$	'U' 'L'	U^TU LL^T	UAU^T L^TAL	$U^{-1}y$ $L^{-T}y$
3	$BAz = \lambda z$	'U' 'L'	U^TU LL^T	UAU^T L^TAL	U^Ty Ly

4 References

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

5 Parameters

1: ITYPE — INTEGER *Input*

On entry: indicates how the standard form is computed as follows:

if ITYPE = 1, then $C = U^{-T}AU^{-1}$ if UPLO = 'U',
$C = L^{-1}AL^{-T}$ if UPLO = 'L';
if ITYPE = 2 or 3, then $C = UAU^T$ if UPLO = 'U',
$C = L^TAL$ if UPLO = 'L'.

Constraint: $1 \leq$ ITYPE ≤ 3.

2: UPLO — CHARACTER*1 *Input*

On entry: indicates whether the upper or lower triangular part of A is stored and how B has been factorized, as follows:

if UPLO = 'U', then the upper triangular part of A is stored and $B = U^T U$;
if UPLO = 'L', then the lower triangular part of A is stored and $B = LL^T$.

Constraint: UPLO = 'U' or 'L'.

3: N — INTEGER *Input*

On entry: n, the order of the matrices A and B.

Constraint: N ≥ 0.

4: AP($*$) — ***real*** array *Input/Output*

Note: the dimension of the array AP must be at least max(1,N$*$(N+1)/2).

On entry: 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 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 AP$(i + (2n-j)(j-1)/2)$ for $i \geq j$.

On exit: the upper or lower triangle of A is overwritten by the corresponding upper or lower triangle of C as specified by ITYPE and UPLO, using the same packed storage format as described above.

5: BP($*$) — ***real*** array *Input*

Note: the dimension of the array B must be at least max(1,N$*$(N+1)/2).

On entry: the Cholesky factor of B as specified by UPLO and returned by F07GDF (SPPTRF/DPPTRF).

6: INFO — INTEGER *Output*

On exit: INFO = 0 unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

INFO < 0

If INFO $= -i$, the ith parameter had an illegal value. An explanatory message is output, and execution of the program is terminated.

7 Accuracy

Forming the reduced matrix C is a stable procedure. However it involves implicit multiplication by B^{-1} if (ITYPE = 1) or B (if ITYPE = 2 or 3). When the routine is used as a step in the computation of eigenvalues and eigenvectors of the original problem, there may be a significant loss of accuracy if B is ill-conditioned with respect to inversion. See the document for F02FDF for further details.

8 Further Comments

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

The complex analogue of this routine is F08TSF (CHPGST/ZHPGST).

9 Example

To compute all the eigenvalues of $Az = \lambda Bz$, where

$$A = \begin{pmatrix} 0.24 & 0.39 & 0.42 & -0.16 \\ 0.39 & -0.11 & 0.79 & 0.63 \\ 0.42 & 0.79 & -0.25 & 0.48 \\ -0.16 & 0.63 & 0.48 & -0.03 \end{pmatrix}$$

and

$$B = \begin{pmatrix} 4.16 & -3.12 & 0.56 & -0.10 \\ -3.12 & 5.03 & -0.83 & 1.18 \\ 0.56 & -0.83 & 0.76 & 0.34 \\ -0.10 & 1.18 & 0.34 & 1.18 \end{pmatrix},$$

using packed storage. Here B is symmetric positive-definite and must first be factorized by F07GDF (SPPTRF/DPPTRF). The program calls F08TEF (SSPGST/DSPGST) to reduce the problem to the standard form $Cy = \lambda y$; then F08GEF (SSPTRD/DSPTRD) to reduce C to tridiagonal form, and F08JFF (SSTERF/DSTERF) to compute the eigenvalues.

9.1 Program Text

Note. The listing of the example program presented below uses bold italicised terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     F08TEF Example Program Text
*     Mark 16 Release. NAG Copyright 1992.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX
      PARAMETER        (NMAX=8)
*     .. Local Scalars ..
      INTEGER          I, INFO, J, N
      CHARACTER        UPLO
*     .. Local Arrays ..
      real             AP(NMAX*(NMAX+1)/2), BP(NMAX*(NMAX+1)/2),
     +                 D(NMAX), E(NMAX-1), TAU(NMAX)
*     .. External Subroutines ..
      EXTERNAL         spptrf, sspgst, ssptrd, ssterf
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08TEF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
*
*        Read A and B from data file
*
         READ (NIN,*) UPLO
         IF (UPLO.EQ.'U') THEN
            READ (NIN,*) ((AP(I+J*(J-1)/2),J=I,N),I=1,N)
            READ (NIN,*) ((BP(I+J*(J-1)/2),J=I,N),I=1,N)
         ELSE IF (UPLO.EQ.'L') THEN
            READ (NIN,*) ((AP(I+(2*N-J)*(J-1)/2),J=1,I),I=1,N)
            READ (NIN,*) ((BP(I+(2*N-J)*(J-1)/2),J=1,I),I=1,N)
         END IF
*
*        Compute the Cholesky factorization of B
*
```

```
         CALL spptrf(UPLO,N,BP,INFO)
*
         WRITE (NOUT,*)
         IF (INFO.GT.0) THEN
            WRITE (NOUT,*) 'B is not positive-definite.'
         ELSE
*
*           Reduce the problem to standard form C*y = lambda*y, storing
*           the result in A
*
            CALL sspgst(1,UPLO,N,AP,BP,INFO)
*
*           Reduce C to tridiagonal form T = (Q**T)*C*Q
*
            CALL ssptrd(UPLO,N,AP,D,E,TAU,INFO)
*
*           Calculate the eigenvalues of T (same as C)
*
            CALL ssterf(N,D,E,INFO)
*
            IF (INFO.GT.0) THEN
               WRITE (NOUT,*) 'Failure to converge.'
            ELSE
*
*              Print eigenvalues
*
               WRITE (NOUT,*) 'Eigenvalues'
               WRITE (NOUT,99999) (D(I),I=1,N)
            END IF
         END IF
      END IF
      STOP
*
99999 FORMAT (3X,(9F8.4))
      END
```

9.2 Program Data

```
F08TEF Example Program Data
  4                            :Value of N
  'L'                          :Value of UPLO
  0.24
  0.39  -0.11
  0.42   0.79  -0.25
 -0.16   0.63   0.48  -0.03   :End of matrix A
  4.16
 -3.12   5.03
  0.56  -0.83   0.76
 -0.10   1.09   0.34   1.18   :End of matrix B
```

9.3 Program Results

```
 F08TEF Example Program Results

 Eigenvalues
    -2.2254 -0.4548  0.1001  1.1270
```