NAG Fortran Library Routine Document

F08SSF (ZHEGST)

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

F08SSF (ZHEGST) reduces a complex Hermitian-definite generalized eigenproblem $Az = \lambda Bz$, $ABz = \lambda z$ or $BAz = \lambda z$ to the standard form $Cy = \lambda y$, where A is a complex Hermitian matrix and B has been factorized by F07FRF (ZPOTRF).

2 Specification

```
SUBROUTINE F08SSF (ITYPE, UPLO, N, A, LDA, B, LDB, INFO)

INTEGER            ITYPE, N, LDA, LDB, INFO
complex*16         A(LDA,*), B(LDB,*)
CHARACTER*1        UPLO
```

The routine may be called by its LAPACK name ***zhegst***.

3 Description

To reduce the complex Hermitian-definite generalized eigenproblem $Az = \lambda Bz$, $ABz = \lambda z$ or $BAz = \lambda z$ to the standard form $Cy = \lambda y$, F08SSF (ZHEGST) must be preceded by a call to F07FRF (ZPOTRF) 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'	$U^H U$ LL^H	$U^{-H} A U^{-1}$ $L^{-1} A L^{-H}$	$U^{-1} y$ $L^{-H} y$
2	$ABz = \lambda z$	'U' 'L'	$U^H U$ LL^H	UAU^H $L^H AL$	$U^{-1} y$ $L^{-H} y$
3	$BAz = \lambda z$	'U' 'L'	$U^H U$ LL^H	UAU^H $L^H AL$	$U^H y$ Ly

4 References

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

5 Parameters

1: ITYPE – INTEGER *Input*

On entry: indicates how the standard form is computed.

ITYPE = 1

if UPLO = 'U', $C = U^{-H}AU^{-1}$;
if UPLO = 'L', $C = L^{-1}AL^{-H}$.

ITYPE = 2 or 3

if UPLO = 'U', $C = UAU^{H}$;
if UPLO = 'L', $C = L^{H}AL$.

Constraint: ITYPE = 1, 2 or 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.

UPLO = 'U'

The upper triangular part of A is stored and $B = U^{H}U$.

UPLO = 'L'

The lower triangular part of A is stored and $B = LL^{H}$.

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

3: N – INTEGER *Input*

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

Constraint: N $\geq$ 0.

4: A(LDA,$*$) – ***complex*16*** array *Input/Output*

Note: the second dimension of the array A must be at least max(1, N).

On entry: the n by n Hermitian 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.

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.

5: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08SSF (ZHEGST) is called.

Constraint: LDA $\geq$ max(1, N).

6: B(LDB,$*$) – ***complex*16*** array *Input*

Note: the second dimension of the array B must be at least max(1, N).

On entry: the Cholesky factor of B as specified by UPLO and returned by F07FRF (ZPOTRF).

7: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F08SSF (ZHEGST) is called.

Constraint: LDB $\geq$ max(1, N).

8: INFO – INTEGER *Output*

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

6 Error Indicators and Warnings

Errors or warnings detected by the routine:

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 F08SSF (ZHEGST) 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 F08SNF (ZHEGV) for further details.

8 Further Comments

The total number of real floating-point operations is approximately $4n^3$.

The real analogue of this routine is F08SEF (DSYGST).

9 Example

This example computes all the eigenvalues of $Az = \lambda Bz$, where

$$A = \begin{pmatrix} -7.36 + 0.00i & 0.77 - 0.43i & -0.64 - 0.92i & 3.01 - 6.97i \\ 0.77 + 0.43i & 3.49 + 0.00i & 2.19 + 4.45i & 1.90 + 3.73i \\ -0.64 + 0.92i & 2.19 - 4.45i & 0.12 + 0.00i & 2.88 - 3.17i \\ 3.01 + 6.97i & 1.90 - 3.73i & 2.88 + 3.17i & -2.54 + 0.00i \end{pmatrix}$$

and

$$B = \begin{pmatrix} 3.23 + 0.00i & 1.51 - 1.92i & 1.90 + 0.84i & 0.42 + 2.50i \\ 1.51 + 1.92i & 3.58 + 0.00i & -0.23 + 1.11i & -1.18 + 1.37i \\ 1.90 - 0.84i & -0.23 - 1.11i & 4.09 + 0.00i & 2.33 - 0.14i \\ 0.42 - 2.50i & -1.18 - 1.37i & 2.33 + 0.14i & 4.29 + 0.00i \end{pmatrix}.$$

Here B is Hermitian positive-definite and must first be factorized by F07FRF (ZPOTRF). The program calls F08SSF (ZHEGST) to reduce the problem to the standard form $Cy = \lambda y$; then F08FSF (ZHETRD) to reduce C to tridiagonal form, and F08JFF (DSTERF) to compute the eigenvalues.

9.1 Program Text

```
*     F08SSF Example Program Text
*     Mark 16 Release. NAG Copyright 1992.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, LDA, LDB, LWORK
      PARAMETER        (NMAX=8,LDA=NMAX,LDB=NMAX,LWORK=64*NMAX)
*     .. Local Scalars ..
      INTEGER          I, INFO, J, N
      CHARACTER        UPLO
*     .. Local Arrays ..
      COMPLEX *16      A(LDA,NMAX), B(LDB,NMAX), TAU(NMAX), WORK(LWORK)
      DOUBLE PRECISION D(NMAX), E(NMAX-1)
*     .. External Subroutines ..
      EXTERNAL         DSTERF, ZHEGST, ZHETRD, ZPOTRF
```

```
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08SSF 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,*) ((A(I,J),J=I,N),I=1,N)
            READ (NIN,*) ((B(I,J),J=I,N),I=1,N)
         ELSE IF (UPLO.EQ.'L') THEN
            READ (NIN,*) ((A(I,J),J=1,I),I=1,N)
            READ (NIN,*) ((B(I,J),J=1,I),I=1,N)
         END IF
*
*        Compute the Cholesky factorization of B
*
         CALL ZPOTRF(UPLO,N,B,LDB,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 ZHEGST(1,UPLO,N,A,LDA,B,LDB,INFO)
*
*           Reduce C to tridiagonal form T = (Q**H)*C*Q
*
            CALL ZHETRD(UPLO,N,A,LDA,D,E,TAU,WORK,LWORK,INFO)
*
*           Calculate the eigenvalues of T (same as C)
*
            CALL DSTERF(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

```
F08SSF Example Program Data
  4                                                         :Value of N
  'L'                                                       :Value of UPLO
 (-7.36, 0.00)
 ( 0.77, 0.43) ( 3.49, 0.00)
 (-0.64, 0.92) ( 2.19,-4.45) ( 0.12, 0.00)
 ( 3.01, 6.97) ( 1.90,-3.73) ( 2.88, 3.17) (-2.54, 0.00)  :End of matrix A
 ( 3.23, 0.00)
 ( 1.51, 1.92) ( 3.58, 0.00)
 ( 1.90,-0.84) (-0.23,-1.11) ( 4.09, 0.00)
 ( 0.42,-2.50) (-1.18,-1.37) ( 2.33, 0.14) ( 4.29, 0.00)  :End of matrix B
```

9.3 Program Results

```
 F08SSF Example Program Results

 Eigenvalues
    -5.9990 -2.9936  0.5047  3.9990
```