NAG Fortran Library Routine Document

F08SPF (ZHEGVX)

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

F08SPF (ZHEGVX) computes selected eigenvalues, and optionally, eigenvectors of a complex generalized Hermitian-definite eigenproblem, of the form

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

where A and B are Hermitian 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 Specification

```
 SUBROUTINE F08SPF (ITYPE, JOBZ, RANGE, UPLO, N, A, LDA, B, LDB, VL, VU,
1                   IL, IU, ABSTOL, M, W, Z, LDZ, WORK, LWORK, RWORK,
2                   IWORK, JFAIL, INFO)
 INTEGER            ITYPE, N, LDA, LDB, IL, IU, M, LDZ, LWORK, IWORK(*),
1                   JFAIL(*), INFO
 double precision   VL, VU, ABSTOL, W(*), RWORK(*)
 complex*16         A(LDA,*), B(LDB,*), Z(LDZ,*), WORK(*)
 CHARACTER*1        JOBZ, RANGE, UPLO
```

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

3 Description

F08SPF (ZHEGVX) first performs a Cholesky factorization of the matrix B as $B = U^H U$, when UPLO = 'U' or $B = LL^H$, 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

$$z^H Bz = I.$$

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

$$z^H B^{-1} z = I.$$

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

1: ITYPE – INTEGER *Input*

On entry: specifies the problem type to be solved:

if ITYPE $= 1$, $Az = \lambda Bz$;
if ITYPE $= 2$, $ABz = \lambda z$;
if ITYPE $= 3$, $BAz = \lambda z$.

2: JOBZ – CHARACTER*1 *Input*

On entry: if JOBZ $=$ 'N', compute eigenvalues only.

If JOBZ $=$ 'V', compute eigenvalues and eigenvectors.

Constraint: JOBZ $=$ 'N' or 'V'.

3: RANGE – CHARACTER*1 *Input*

On entry: if RANGE $=$ 'A', all eigenvalues will be found.

If RANGE $=$ 'V', all eigenvalues in the half-open interval $(\mathrm{VL}, \mathrm{VU}]$ will be found.

If RANGE $=$ 'I', the ILth to IUth eigenvalues will be found.

4: UPLO – CHARACTER*1 *Input*

On entry: if UPLO $=$ 'U', the upper triangles of A and B are stored.

If UPLO $=$ 'L', the lower triangles of A and B are stored.

5: N – INTEGER *Input*

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

Constraint: $\mathrm{N} \geq 0$.

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

Note: the second dimension of the array A must be at least $\max(1, \mathrm{N})$.

On entry: the n by n Hermitian matrix A.

If UPLO $=$ 'U', the leading n by n upper triangular part of A contains the upper triangular part of the matrix A.

If UPLO $=$ 'L', the leading n by n lower triangular part of A contains the lower triangular part of the matrix A.

On exit: the lower triangle (if UPLO $=$ 'L') or the upper triangle (if UPLO $=$ 'U') of A, including the diagonal, is destroyed.

7: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08SPF (ZHEGVX) is called.

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

8: B(LDB,$*$) – ***complex*16*** array *Input/Output*

Note: the second dimension of the array B must be at least $\max(1, \mathrm{N})$.

On entry: the Hermitian 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.

On exit: if INFO $\leq$ N, the part of B containing the matrix is overwritten by the triangular factor U or L from the Cholesky factorization $\mathrm{B} = U^H U$ or $\mathrm{B} = LL^H$.

9: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F08SPF (ZHEGVX) is called.

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

10: VL – ***double precision*** *Input*
11: VU – ***double precision*** *Input*

On entry: if RANGE = 'V', the lower and upper bounds of the interval to be searched for eigenvalues.

Constraint: VL $<$ VU.

If RANGE = 'A' or 'I', VL and VU are not referenced.

12: IL – INTEGER *Input*
13: IU – INTEGER *Input*

On entry: 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.

14: ABSTOL – ***double precision*** *Input*

On entry: 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

$$\mathrm{ABSTOL} + \epsilon \max(|a|, |b|),$$

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 $2 \times \mathrm{X02AMF}()$, not zero. If this routine returns with INFO > 0, indicating that some eigenvectors did not converge, try setting ABSTOL to $2 \times \mathrm{X02AMF}()$. See Demmel and Kahan (1990).

15: M – INTEGER *Output*

On exit: the total number of eigenvalues found.

If RANGE = 'A', M = N.

If RANGE = 'I', M = IU $-$ IL $+ 1$.

Constraint: $0 \leq$ M $\leq$ N.

16: W$(*)$ – ***double precision*** array *Output*

Note: the dimension of the array W must be at least $\max(1, \mathrm{N})$.

On exit: the first M elements contain the selected eigenvalues in ascending order.

17: Z(LDZ,$*$) – ***complex*16*** array *Output*

Note: the second dimension of the array Z must be at least $\max(1, \mathrm{M})$.

On exit: if JOBZ = 'V', then if INFO = 0, the first m columns of Z contain the orthonormal eigenvectors of the matrix A corresponding to the selected eigenvalues, with the ith column of Z holding the eigenvector associated with $\mathrm{W}(i)$. The eigenvectors are normalised as follows:

if ITYPE = 1 or 2, $Z^T BZ = I$;
if ITYPE = 3, $Z^T B^{-1} Z = I$.

If JOBZ = 'N', Z is not referenced.

If an eigenvector fails to converge, then that column of Z contains the latest approximation to the eigenvector, and the index of the eigenvector is returned in JFAIL.

Note: the user must ensure that at least $\max(1, \mathrm{M})$ columns are supplied in the array Z; if RANGE = 'V', the exact value of M is not known in advance and an upper bound must be used.

18: LDZ – INTEGER *Input*

On entry: the first dimension of the array Z as declared in the (sub)program from which F08SPF (ZHEGVX) is called.

Constraints:

if JOBZ = 'V', LDZ $\geq \max(1, \mathrm{N})$;
LDZ ≥ 1 otherwise.

19: WORK($*$) – ***complex*16*** array *Workspace*

Note: the dimension of the array WORK must be at least $\max(1, \mathrm{LWORK})$.

On exit: if INFO = 0, $\mathrm{WORK}(1)$ returns the optimal LWORK.

20: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08SPF (ZHEGVX) is called.

For optimal efficiency, LWORK $\geq (nb + 1) \times \mathrm{N}$, where nb is the optimal block size for F08FSF (ZHETRD).

If LWORK $= -1$, a workspace query is assumed; the routine only calculates the optimal size of the WORK array, returns this value as the first entry of the WORK array, and no error message related to LWORK is issued.

Constraint: LWORK $\geq \max(1, 2 \times \mathrm{N} - 1)$.

21: RWORK($*$) – ***double precision*** array *Workspace*

Note: the dimension of the array RWORK must be at least $\max(1, 7 \times \mathrm{N})$.

22: IWORK($*$) – INTEGER array *Workspace*

Note: the dimension of the array IWORK must be at least $\max(1, 5 \times \mathrm{N})$.

23: JFAIL($*$) – INTEGER array *Output*

Note: the dimension of the array JFAIL must be at least $\max(1, \mathrm{N})$.

On exit: if JOBZ = 'V', then if INFO = 0, the first M elements of JFAIL are zero. If INFO > 0, JFAIL contains the indices of the eigenvectors that failed to converge.

If JOBZ = 'N', JFAIL is not referenced.

24: 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 argument had an illegal value.

INFO > 0

F07FRF (ZPOTRF) or F08FPF (ZHEEVX) returned an error code:

$\leq$ N if INFO $= i$, F08FPF (ZHEEVX) failed to converge; i eigenvectors failed to converge. Their indices are stored in array JFAIL;

$>$ N if INFO $=$ N $+ i$, for $1 \leq i \leq$ 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.

8 Further Comments

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

The real analogue of this routine is F08SBF (DSYGVX).

9 Example

To find all the eigenvalues in the half-open interval $(-3, 3)$, and corresponding eigenvectors, of the generalized Hermitian eigenproblem $Az = \lambda Bz$, where

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

and

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

The example program for F08SQF (ZHEGVD) illustrates solving a generalized Hermitian eigenproblem of the form $ABz = \lambda z$.

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.

```
*     F08SPF Example Program Text
*     Mark 21.  NAG Copyright 2004.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          MMAX, NB, NMAX
      PARAMETER        (MMAX=5,NB=64,NMAX=10)
      INTEGER          LDA, LDB, LDZ, LWORK
      PARAMETER        (LDA=NMAX,LDB=NMAX,LDZ=NMAX,LWORK=(NB+1)*NMAX)
      DOUBLE PRECISION ZERO
      PARAMETER        (ZERO=0.0D+0)
*     .. Local Scalars ..
      DOUBLE PRECISION ABSTOL, VL, VU
      INTEGER          I, IFAIL, IL, INFO, IU, J, LWKOPT, M, N
*     .. Local Arrays ..
      COMPLEX *16      A(LDA,NMAX), B(LDB,NMAX), WORK(LWORK),
     +                 Z(LDZ,MMAX)
      DOUBLE PRECISION RWORK(7*NMAX), W(NMAX)
      INTEGER          INDEX(NMAX), IWORK(5*NMAX)
*     .. External Subroutines ..
      EXTERNAL         X04DAF, ZHEGVX
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08SPF Example Program Results'
      WRITE (NOUT,*)
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
*
*        Read the lower and upper bounds of the interval to be searched,
*        and read the upper triangular parts of the matrices A and B
*
         READ (NIN,*) VL, VU
         READ (NIN,*) ((A(I,J),J=I,N),I=1,N)
         READ (NIN,*) ((B(I,J),J=I,N),I=1,N)
*
*        Set the absolute error tolerance for eigenvalues.  With ABSTOL
*        set to zero, the default value is used instead
*
         ABSTOL = ZERO
*
*        Solve the generalized Hermitian eigenvalue problem
*        A*x = lambda*B*x (ITYPE = 1)
*
         CALL ZHEGVX(1,'Vectors','Values in range','Upper',N,A,LDA,B,
     +              LDB,VL,VU,IL,IU,ABSTOL,M,W,Z,LDZ,WORK,LWORK,RWORK,
     +              IWORK,INDEX,INFO)
         LWKOPT = WORK(1)
*
         IF (INFO.GE.0 .AND. INFO.LE.N .AND. M.LE.MMAX) THEN
*
*           Print solution
*
            WRITE (NOUT,99999) 'Number of eigenvalues found =', M
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Eigenvalues'
            WRITE (NOUT,99998) (W(J),J=1,M)
*
            IFAIL = 0
            CALL X04DAF('General',' ',N,M,Z,LDZ,'Selected eigenvectors',
     +                  IFAIL)
            IF (INFO.GT.0) THEN
               WRITE (NOUT,99999)
     +           'INFO eigenvectors failed to converge, INFO =', INFO
               WRITE (NOUT,*)
```

```
     +           'Indices of eigenvectors that did not converge'
                 WRITE (NOUT,99997) (INDEX(J),J=1,M)
              END IF
          ELSE IF (INFO.GT.N .AND. INFO.LE.2*N) THEN
              I = INFO - N
              WRITE (NOUT,99996) 'The leading minor of order ', I,
     +          ' of B is not positive definite'
          ELSE IF (M.GT.MMAX) THEN
              WRITE (NOUT,99995) 'M greater than MMAX, M =', M,
     +          ', MMAX =', MMAX
          ELSE
              WRITE (NOUT,99999) 'Failure in ZHEGVX. INFO =', INFO
          END IF
*
*         Print workspace information
*
          WRITE (NOUT,*)
          IF (LWORK.LT.LWKOPT) THEN
              WRITE (NOUT,99994) 'Optimum complex workspace required = ',
     +          LWKOPT, 'Workspace provided in WORK         = ', LWORK
          END IF
       ELSE
          WRITE (NOUT,*)
          WRITE (NOUT,*) 'NMAX too small'
       END IF
       STOP
*
99999 FORMAT (1X,A,I5)
99998 FORMAT (3X,(8F8.4))
99997 FORMAT (3X,(8I8))
99996 FORMAT (1X,A,I4,A)
99995 FORMAT (1X,A,I5,A,I5)
99994 FORMAT (1X,A,I5,/1X,A,I5)
      END
```

9.2 Program Data

```
F08SPF Example Program Data

   4                                                          :Value of N

  -3.0           3.0                                         :Values of VL and VU

 (-7.36, 0.00) ( 0.77, -0.43) (-0.64, -0.92) ( 3.01, -6.97)
               ( 3.49,  0.00) ( 2.19,  4.45) ( 1.90,  3.73)
                              ( 0.12,  0.00) ( 2.88, -3.17)
                                             (-2.54,  0.00) :End of matrix A

 ( 3.23, 0.00) ( 1.51, -1.92) ( 1.90,  0.84) ( 0.42,  2.50)
               ( 3.58,  0.00) (-0.23,  1.11) (-1.18,  1.37)
                              ( 4.09,  0.00) ( 2.33, -0.14)
                                             ( 4.29,  0.00) :End of matrix B
```

9.3 Program Results

```
 F08SPF Example Program Results

 Number of eigenvalues found =    2

 Eigenvalues
    -2.9936  0.5047
 Selected eigenvectors
          1       2
 1  -0.3504  0.2835
     0.6060 -0.5806

 2  -0.0993 -0.3769
     0.0631 -0.3194

 3   0.6851 -0.3338
    -0.5987 -0.0134

 4  -0.8127  0.6663
     0.0000  0.0000
```