F02HCF – 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

F02HCF computes selected eigenvalues, and optionally the corresponding eigenvectors, of a complex Hermitian matrix.

2 Specification

```
 SUBROUTINE F02HCF(JOB, RANGE, UPLO, N, A, LDA, WL, WU, IL, IU,
1                  MEST, M, W, Z, LDZ, WORK, LWORK, RWORK, IWORK,
2                  IFAIL)
 INTEGER           N, LDA, IL, IU, MEST, M, LDZ, LWORK, IWORK(*),
1                  IFAIL
 real              WL, WU, W(*), RWORK(*)
 complex           A(LDA,*), Z(LDZ,MEST), WORK(LWORK)
 CHARACTER*1       JOB, RANGE, UPLO
```

3 Description

This routine computes selected eigenvalues, and optionally the corresponding eigenvectors, of a complex Hermitian matrix A:

$$Az_i = \lambda_i z_i, \quad \text{with real } \lambda_i.$$

The eigenvalues λ_i are selected either by *value* (all the eigenvalues in a half-open interval):

$$w_l \leq \lambda_i < w_u$$

or by *index*, assuming that the eigenvalues are indexed in *ascending* order:

$$i_l \leq i \leq i_u, \text{ where } \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n.$$

4 References

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

[2] Parlett B N (1980) *The Symmetric Eigenvalue Problem* Prentice–Hall

5 Parameters

1: JOB — CHARACTER*1 *Input*

On entry: indicates whether eigenvectors are to be computed as follows:

if JOB = 'N', then only eigenvalues are computed;
if JOB = 'V', then eigenvalues and eigenvectors are computed.

Constraint: JOB = 'N' or 'V'.

2: RANGE — CHARACTER*1 *Input*

On entry: indicates how eigenvalues are to be selected, as follows:

if RANGE = 'V', then eigenvalues are selected by value (see WL and WU);
if RANGE = 'I', then eigenvalues are selected by index (see IL and IU).

Constraint: RANGE = 'V' or 'I'.

3: UPLO — CHARACTER*1 *Input*

On entry: indicates whether the upper or lower triangular part of A is stored as follows:

if UPLO = 'U', then the upper triangular part of A is stored;
if UPLO = 'L', then the lower triangular part of A is stored.

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

4: N — INTEGER *Input*

On entry: n, the order of the matrix A.

Constraint: N $\geq$ 0.

5: A(LDA,$*$) — ***complex*** 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 triangle of A must be stored and the elements of the array below the diagonal need not be set; if UPLO = 'L', the lower triangle of A must be stored and the elements of the array above the diagonal need not be set.

On exit: the contents of A are overwritten. The diagonal and first off-diagonal contain the upper or lower triangle of the real symmetric tridiagonal matrix T (see Section 8).

6: LDA — INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F02HCF is called.

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

7: WL — ***real*** *Input*

8: WU — ***real*** *Input*

On entry: w_l and w_u, the lower and upper bounds of the interval in which eigenvalues are selected if RANGE = 'V'. Not referenced if RANGE = 'I'.

Constraint: WU > WL.

9: IL — INTEGER *Input*

10: IU — INTEGER *Input*

On entry: i_l and i_u, the lower and upper bounds of the indices of the eigenvalues which are selected if RANGE = 'I'. Not referenced if RANGE = 'V'.

Constraint: 1 $\leq$ IL $\leq$ IU $\leq$ N, if N > 0.

11: MEST — INTEGER *Input*

On entry: the second dimension of the array Z as declared in the (sub)program from which F02HCF is called. If JOB = 'V', MEST must be an upper bound on m, the number of eigenvalues and eigenvectors selected. No eigenvectors are computed if MEST < m. MEST is not referenced if JOB = 'N'.

Constraint: MEST $\geq$ max(1,m); MEST $\geq$ IU $-$ IL + 1 if RANGE = 'I'.

12: M — INTEGER *Output*

On exit: m, the number of eigenvalues actually selected. If RANGE = 'I', $m = i_u - i_l + 1$.

13: W($*$) — ***real*** array *Output*

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

On exit: the first M elements hold the selected eigenvalues in ascending order; elements M + 1 to N are used as workspace.

14: Z(LDZ,MEST) — ***complex*** array *Output*

On exit: if JOB = 'V', the first M columns of Z contain the selected eigenvectors, with the ith column holding the eigenvector z_i associated with the eigenvalue λ_i (stored in W(i)). Z is not referenced if JOB = 'N'.

15: LDZ — INTEGER *Input*

On entry: the first dimension of the array Z as declared in the (sub)program from which F02HCF is called.

Constraint: LDZ $\geq$ max(1,N) if JOB = 'V'; LDZ $\geq$ 1 otherwise.

16: WORK(LWORK) — ***complex*** array *Workspace*

17: LWORK — INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F02HCF is called. On some high-performance computers, increasing the dimension of WORK will enable the routine to run faster; a value of $64 \times$ N should allow near-optimal performance on almost all machines.

Constraint: LWORK $\geq$ max(1,2$\times$N).

18: RWORK(*) — ***real*** array *Workspace*

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

19: IWORK(*) — INTEGER array *Workspace*

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

20: IFAIL — INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

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

6 Error Indicators and Warnings

If on entry IFAIL = 0 or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors detected by the routine:

IFAIL = 1

On entry, JOB $\neq$ 'N' or 'V',
or RANGE $\neq$ 'V' or 'I',
or UPLO $\neq$ 'U' or 'L',
or N $<$ 0,
or LDA $<$ max(1,N),
or WU $\leq$ WL when RANGE = 'V',
or IL $<$ 1 when RANGE = 'I',
or IU $>$ N, or IL $>$ IU and N $>$ 0, when RANGE = 'I',
or MEST $<$ 1,
or LDZ $<$ 1, or LDZ $<$ N when JOB = 'V',
or LWORK $<$ max(1,2$\times$N).

IFAIL = 2

The bisection algorithm failed to compute all the eigenvalues. No eigenvectors have been computed.

IFAIL = 3

There are more than MEST eigenvalues in the specified range. The actual number of eigenvalues in the range is returned in M. No eigenvectors have been computed. Rerun with the second dimension of Z= MEST $\geq$ M.

IFAIL = 4

Inverse iteration failed to compute all the specified eigenvectors. If an eigenvector failed to converge, the corresponding column of Z is set to zero.

IFAIL = 5

For some i, A(i,i) has a non-zero imaginary part (thus A is not Hermitian).

7 Accuracy

If λ_i is an exact eigenvalue, and $\tilde{\lambda}_i$ is the corresponding computed value, then

$$|\tilde{\lambda}_i - \lambda_i| \leq c(n)\epsilon\|A\|_2,$$

where $c(n)$ is a modestly increasing function of n, and ϵ is the ***machine precision***.

If z_i is the corresponding exact eigenvector, and $\tilde{z}_i$ is the corresponding computed eigenvector, then the angle $\theta(\tilde{z}_i, z_i)$ between them is bounded as follows:

$$\theta(\tilde{z}_i, z_i) \leq \frac{c(n)\epsilon\|A\|_2}{\min\limits_{i\neq j}|\lambda_i - \lambda_j|}.$$

Thus the accuracy of a computed eigenvector depends on the gap between its eigenvalue and all the other eigenvalues.

8 Further Comments

The routine calls routines from LAPACK in Chapter F08. It first reduces A to real symmetric tridiagonal form T, using a unitary similarity transformation: $A = QTQ^H$. Eigenvalues of T are computed by bisection. If eigenvectors are required, eigenvectors of T are computed by inverse iteration, and are transformed to eigenvectors of A by premultiplying them with the unitary matrix Q that was used in the reduction to tridiagonal form.

Each eigenvector z is normalized so that $\|z\|_2 = 1$ and the element of largest absolute value is real and positive.

The time taken by the routine is approximately proportional to n^3.

The routine can be used to compute *all* eigenvalues, and optionally *all* eigenvectors, by setting RANGE = 'I', IL = 1 and IU = N. In some circumstances it may do this more efficiently than F02HAF, but this depends on the machine, the size of the problem, and the distribution of eigenvalues. Eigenvectors computed by F02HCF may not be orthogonal to as high a degree of accuracy as those computed by F02HAF.

9 Example

To compute the two smallest eigenvalues of the matrix A, and their corresponding eigenvectors, where

$$A = \begin{pmatrix} -2.28+0.00i & 1.78-2.03i & 2.26+0.10i & -0.12+2.53i \\ 1.78+2.03i & -1.12+0.00i & 0.01+0.43i & -1.07+0.86i \\ 2.26-0.10i & 0.01-0.43i & -0.37+0.00i & 2.31-0.92i \\ -0.12-2.53i & -1.07-0.86i & 2.31+0.92i & -0.73+0.00i \end{pmatrix}.$$

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.

```
*     F02HCF Example Program Text
*     Mark 17 Release. NAG Copyright 1995.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, MMAX, LDA, LDZ, LWORK
      PARAMETER        (NMAX=8,MMAX=3,LDA=NMAX,LDZ=NMAX,LWORK=64*NMAX)
*     .. Local Scalars ..
      real             WL, WU
      INTEGER          I, IFAIL, IL, IU, J, M, N
      CHARACTER        UPLO
*     .. Local Arrays ..
      complex          A(LDA,NMAX), WORK(LWORK), Z(LDZ,MMAX)
      real             RWORK(7*NMAX), W(NMAX)
      INTEGER          IWORK(5*NMAX)
      CHARACTER        CLABS(1), RLABS(1)
*     .. External Subroutines ..
      EXTERNAL         F02HCF, X04DBF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F02HCF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, IL, IU
      IF (N.LE.NMAX) THEN
*
*        Read A from data file
*
         READ (NIN,*) UPLO
         IF (UPLO.EQ.'U') THEN
            READ (NIN,*) ((A(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)
         END IF
*
*        Compute selected eigenvalues and eigenvectors
*
         IFAIL = 0
*
         CALL F02HCF('Vectors','Index',UPLO,N,A,LDA,WL,WU,IL,IU,MMAX,M,
     +               W,Z,LDZ,WORK,LWORK,RWORK,IWORK,IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Eigenvalues'
         WRITE (NOUT,99999) (W(I),I=1,M)
         WRITE (NOUT,*)
*
         CALL X04DBF('General',' ',N,M,Z,LDZ,'Bracketed','F7.4',
     +               'Eigenvectors','Integer',RLABS,'Integer',CLABS,80,
     +               0,IFAIL)
*
      END IF
      STOP
*
```

```
99999 FORMAT (3X,4(F12.4,6X))
      END
```

9.2 Program Data

```
F02HCF Example Program Data
  4  1  2                                                      :Values of N, IL, IU
  'L'                                                          :Value of UPLO
 (-2.28, 0.00)
 ( 1.78, 2.03) (-1.12, 0.00)
 ( 2.26,-0.10) ( 0.01,-0.43) (-0.37, 0.00)
 (-0.12,-2.53) (-1.07,-0.86) ( 2.31, 0.92) (-0.73, 0.00)   :End of matrix A
```

9.3 Program Results

```
F02HCF Example Program Results

Eigenvalues
       -6.0002          -3.0030

Eigenvectors
                   1                 2
1  ( 0.7299, 0.0000) (-0.2120, 0.1497)
2  (-0.1663,-0.2061) ( 0.7307, 0.0000)
3  (-0.4165,-0.1417) (-0.3291, 0.0479)
4  ( 0.1743, 0.4162) ( 0.5200, 0.1329)
```