NAG Fortran Library Routine Document

F02ECF

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

F02ECF computes selected eigenvalues and eigenvectors of a real general matrix.

2 Specification

```
 SUBROUTINE F02ECF(CRIT, N, A, LDA, WL, WU, MEST, M, WR, WI, VR, LDVR,
1                  VI, LDVI, WORK, LWORK, IWORK, BWORK, IFAIL)
 INTEGER           N, LDA, MEST, M, LDVR, LDVI, LWORK, IWORK(*), IFAIL
 real              A(LDA,*), WL, WU, WR(*), WI(*), VR(LDVR,MEST),
1                  VI(LDVI,MEST), WORK(LWORK)
 LOGICAL           BWORK(*)
 CHARACTER*1       CRIT
```

3 Description

This routine computes selected eigenvalues and the corresponding right eigenvectors of a real general matrix A:

$$Ax_i = \lambda_i x_i.$$

Eigenvalues λ_i may be selected either by *modulus*, satisfying:

$$w_l \leq |\lambda_i| \leq w_u,$$

or by *real part*, satisfying:

$$w_l \leq \mathrm{Re}(\lambda_i) \leq w_u.$$

Note that even though A is real, λ_i and x_i may be complex. If x_i is an eigenvector corresponding to a complex eigenvalue λ_i, then the complex conjugate vector $\bar{x}_i$ is the eigenvector corresponding to the complex conjugate eigenvalue $\bar{\lambda}_i$. The eigenvalues in a complex conjugate pair λ_i and $\bar{\lambda}_i$ are either both selected or both not selected.

4 References

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

5 Parameters

1: CRIT – CHARACTER*1 *Input*

On entry: indicates the criterion for selecting eigenvalues:

if CRIT = 'M', then eigenvalues are selected according to their moduli: $w_l \leq |\lambda_i| \leq w_u$;

if CRIT = 'R', then eigenvalues are selected according to their real parts: $w_l \leq \mathrm{Re}(\lambda_i) \leq w_u$.

Constraint: CRIT = 'M' or 'R'.

2: N – INTEGER *Input*

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

Constraint: N ≥ 0.

3: A(LDA,*) – ***real*** 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 general matrix A.

On exit: A contains the Hessenberg form of the balanced input matrix A' (see Section 8).

4: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F02ECF is called.

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

5: WL – ***real*** *Input*
6: WU – ***real*** *Input*

On entry: w_l and w_u, the lower and upper bounds on the criterion for the selected eigenvalues (see CRIT).

Constraint: WU $>$ WL.

7: MEST – INTEGER *Input*

On entry: the second dimension of the arrays VR and VI as declared in the (sub)program from which F02ECF is called. MEST must be an upper bound on m, the number of eigenvalues and eigenvectors selected. No eigenvectors are computed if MEST $< m$.

Constraint: MEST $\geq \max(1, m)$.

8: M – INTEGER *Output*

On exit: m, the number of eigenvalues actually selected.

9: WR($*$) – ***real*** array *Output*
10: WI($*$) – ***real*** array *Output*

Note: the dimensions of the arrays WR and WI must each be at least $\max(1, \mathrm{N})$.

On exit: the first M elements of WR and WI hold the real and imaginary parts, respectively, of the selected eigenvalues; elements $\mathrm{M} + 1$ to N contain the other eigenvalues. Complex conjugate pairs of eigenvalues are stored in consecutive elements of the arrays, with the eigenvalue having positive imaginary part first. See also Section 8.

11: VR(LDVR,MEST) – ***real*** array *Output*

On exit: VR contains the real parts of the selected eigenvectors, with the ith column holding the real part of the eigenvector associated with the eigenvalue λ_i (stored in WR(i) and WI(i)).

12: LDVR – INTEGER *Input*

On entry: the first dimension of the array VR as declared in the (sub)program from which F02ECF is called.

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

13: VI(LDVI,MEST) – ***real*** array *Output*

On exit: VI contains the imaginary parts of the selected eigenvectors, with the ith column holding the imaginary part of the eigenvector associated with the eigenvalue λ_i (stored in WR(i) and WI(i)).

14: LDVI – INTEGER *Input*

On entry: the first dimension of the array VI as declared in the (sub)program from which F02ECF is called.

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

15: WORK(LWORK) – ***real*** array *Workspace*

16: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F02ECF is called.

Constraint: LWORK $\geq \max(1, N \times (N+4))$.

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

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

18: BWORK($*$) – LOGICAL array *Workspace*

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

19: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

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

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, for users not familiar with this parameter the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

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 or warnings detected by the routine:

IFAIL $= 1$

On entry, CRIT $\neq$ 'M' or 'R',
or N < 0,
or LDA $< \max(1, N)$,
or WU $\leq$ WL,
or MEST < 1,
or LDVR $< \max(1, N)$,
or LDVI $< \max(1, N)$,
or LWORK $< \max(1, N \times (N+4))$.

IFAIL $= 2$

The QR 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 VR and VI $=$ MEST $\geq$ M.

IFAIL $= 4$

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

7 Accuracy

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

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

where $c(n)$ is a modestly increasing function of n, ϵ is the ***machine precision***, and s_i is the reciprocal condition number of λ_i; A' is the balanced form of the original matrix A (see Section 8), and $\|A'\| \leq \|A\|$.

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

$$\theta(\tilde{x}_i, x_i) \leq \frac{c(n)\epsilon\|A'\|_2}{sep_i},$$

where sep_i is the reciprocal condition number of x_i.

The condition numbers s_i and sep_i may be computed from the Hessenberg form of the balanced matrix A' which is returned in the array A. This requires calling F08PEF (SHSEQR/DHSEQR) with JOB = 'S' to compute the Schur form of A', followed by F08QLF (STRSNA/DTRSNA).

8 Further Comments

The routine calls routines from LAPACK in Chapter F08. It first balances the matrix, using a diagonal similarity transformation to reduce its norm; and then reduces the balanced matrix A' to upper Hessenberg form H, using an orthogonal similarity transformation: $A' = QHQ^T$. The routine uses the Hessenberg QR algorithm to compute all the eigenvalues of H, which are the same as the eigenvalues of A. It computes the eigenvectors of H which correspond to the selected eigenvalues, using inverse iteration. It premultiplies the eigenvectors by Q to form the eigenvectors of A'; and finally transforms the eigenvectors to those of the original matrix A.

Each eigenvector x (real or complex) is normalized so that $\|x\|_2 = 1$, and the element of largest absolute value is real and positive.

The inverse iteration routine may make a small perturbation to the real parts of close eigenvalues, and this may shift their moduli just outside the specified bounds. If you are relying on eigenvalues being within the bounds, you should test them on return from F02ECF.

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

The routine can be used to compute *all* eigenvalues and eigenvectors, by setting WL large and negative, and WU large and positive. In some circumstances it may do this more efficiently than F02EBF, but this depends on the machine, the size of the problem, and the distribution of eigenvalues.

9 Example

To compute those eigenvalues of the matrix A whose moduli lie in the range [0.2,0.5], and their corresponding eigenvectors, where

$$A = \begin{pmatrix} 0.35 & 0.45 & -0.14 & -0.17 \\ 0.09 & 0.07 & -0.54 & 0.35 \\ -0.44 & -0.33 & -0.03 & 0.17 \\ 0.25 & -0.32 & -0.13 & 0.11 \end{pmatrix}.$$

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.

```
*     F02ECF Example Program Text
*     Mark 17 Release. NAG Copyright 1995.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, MMAX, LDA, LDV, LDVI, LDVR, LWORK
      PARAMETER        (NMAX=8,MMAX=3,LDA=NMAX,LDV=NMAX,LDVI=NMAX,
     +                 LDVR=NMAX,LWORK=64*NMAX)
*     .. Local Scalars ..
      real             WL, WU
      INTEGER          I, IFAIL, J, M, N
*     .. Local Arrays ..
      complex          V(LDV,NMAX)
      real             A(LDA,NMAX), VI(LDVI,MMAX), VR(LDVR,MMAX),
     +                 WI(NMAX), WORK(LWORK), WR(NMAX)
      INTEGER          IWORK(NMAX)
      LOGICAL          BWORK(NMAX)
      CHARACTER        CLABS(1), RLABS(1)
*     .. External Subroutines ..
      EXTERNAL         F02ECF, X04DBF
*     .. Intrinsic Functions ..
      INTRINSIC        cmplx
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F02ECF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, WL, WU
      IF (N.LE.NMAX) THEN
*
*        Read A from data file
*
         READ (NIN,*) ((A(I,J),J=1,N),I=1,N)
*
*        Compute selected eigenvalues and eigenvectors of A
*
         IFAIL = 0
*
         CALL F02ECF('Moduli',N,A,LDA,WL,WU,MMAX,M,WR,WI,VR,LDVR,VI,
     +               LDVI,WORK,LWORK,IWORK,BWORK,IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Eigenvalues'
         WRITE (NOUT,99999) (' (',WR(I),',',WI(I),')',I=1,M)
         WRITE (NOUT,*)
         DO 40 J = 1, M
            DO 20 I = 1, N
               V(I,J) = cmplx(VR(I,J),VI(I,J))
   20       CONTINUE
   40    CONTINUE
*
         CALL X04DBF('General',' ',N,M,V,LDV,'Bracketed','F7.4',
     +               'Eigenvectors','Integer',RLABS,'Integer',CLABS,80,
     +               0,IFAIL)
*
      END IF
      STOP
*
99999 FORMAT ((3X,4(A,F7.4,A,F7.4,A,:)))
      END
```

9.2 Program Data

```
F02ECF Example Program Data
  4  0.2  0.5                    :Values of N, WL, WU
  0.35   0.45  -0.14  -0.17
  0.09   0.07  -0.54   0.35
 -0.44  -0.33  -0.03   0.17
  0.25  -0.32  -0.13   0.11   :End of matrix A
```

9.3 Program Results

```
 F02ECF Example Program Results

 Eigenvalues
    (-0.0994, 0.4008) (-0.0994,-0.4008)

 Eigenvectors
                    1                  2
 1  (-0.1933, 0.2546) (-0.1933,-0.2546)
 2  ( 0.2519,-0.5224) ( 0.2519, 0.5224)
 3  ( 0.0972,-0.3084) ( 0.0972, 0.3084)
 4  ( 0.6760, 0.0000) ( 0.6760,-0.0000)
```