F02BCF – 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

F02BCF calculates selected eigenvalues and eigenvectors of a real nonsymmetric matrix by reduction to Hessenberg form, the QR algorithm and inverse iteration, where the moduli of the selected eigenvalues lie between two given values.

2 Specification

```
      SUBROUTINE F02BCF(A, IA, N, ALB, UB, M, MM, RR, RI, VR, IVR, VI,
     1                  IVI, INTGER, ICNT, C, B, IB, U, V, IFAIL)
      INTEGER           IA, N, M, MM, IVR, IVI, INTGER(N), ICNT(N), IB,
     1                  IFAIL
      real              A(IA,N), ALB, UB, RR(N), RI(N), VR(IVR,M),
     1                  VI(IVI,M), B(IB,N), U(N), V(N)
      LOGICAL           C(N)
```

3 Description

The matrix A is first reduced to upper Hessenberg form H using stabilised elementary similarity transformations. All the eigenvalues are then found using the QR algorithm for real Hessenberg matrices. Selected eigenvectors of H, corresponding to the eigenvalues whose moduli lie in the required range, are then found using inverse iteration. The eigenvectors are then transformed back to those of the original matrix, A.

4 References

[1] Wilkinson J H and Reinsch C (1971) *Handbook for Automatic Computation II, Linear Algebra* Springer-Verlag

5 Parameters

1: A(IA,N) — ***real*** array *Input/Output*

On entry: the n by n matrix A.

On exit: the derived upper Hessenberg matrix H, together with information about multipliers used in the reduction.

2: IA — INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F02BCF is called.

Constraint: IA $\geq$ N.

3: N — INTEGER *Input*

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

4: ALB — ***real*** *Input*

5: UB — ***real*** *Input*

On entry: the lower and upper bounds of the interval specified for selecting eigenvalues. Eigenvalues whose moduli lie within the interval are selected.

6: M — INTEGER *Input*

On entry: an upper bound for the number of selected eigenvalues.

7: MM — INTEGER *Output*

On exit: the actual number of selected eigenvalues.

8: RR(N) — ***real*** array *Output*

9: RI(N) — ***real*** array *Output*

On exit: the real and imaginary parts respectively, of the eigenvalues. The first MM elements of each array contain the selected eigenvalues and the remaining elements the other eigenvalues. See also Section 8.

10: VR(IVR,M) — ***real*** array *Output*

On exit: the real parts of the MM selected eigenvectors, stored by columns. The ith column corresponds to the ith eigenvalue. The eigenvectors are normalised such that the sum of squares of the moduli of the elements is equal to 1 and the element of largest modulus is real.

11: IVR — INTEGER *Input*

On entry: the first dimension of the array VR as declared in the (sub)program from which F02BCF is called.

Constraint: IVR $\geq$ N.

12: VI(IVI,M) — ***real*** array *Output*

On exit: the imaginary parts of the MM selected eigenvectors, stored by columns. The ith column corresponds to the ith eigenvalue.

13: IVI — INTEGER *Input*

On entry: the first dimension of the array VI as declared in the (sub)program from which F02BCF is called.

Constraint: IVI $\geq$ N.

14: INTGER(N) — INTEGER array *Output*

On exit: details of the row and column interchanges involved in the reduction of the original matrix to upper Hessenberg form.

15: ICNT(N) — INTEGER array *Workspace*

16: C(N) — LOGICAL array *Workspace*

17: B(IB,N) — ***real*** array *Workspace*

18: IB — INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F02BCF is called.

Constraint: IB $\geq$ N + 2.

19: U(N) — ***real*** array *Workspace*

20: V(N) — ***real*** array *Workspace*

21: 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

Errors detected by the routine:

IFAIL = 1

More than 30 × N iterations are required to isolate all the eigenvalues.

IFAIL = 2

There are no eigenvalues whose moduli lie between ALB and UB. The number of selected eigenvalues MM will be set to zero.

IFAIL = 3

M is less than the number of eigenvalues in the given range. On exit, MM contains the number of eigenvalues in the range. Rerun with this value for M.

7 Accuracy

The accuracy of the results depends on the original matrix and the multiplicity of the roots. For a detailed error analysis see Wilkinson and Reinsch [1] pp 352, 367 and 436.

8 Further Comments

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

This routine should only be used when less than 25% of the eigenvectors are required. If this is not the case, it is more efficient to determine the complete eigensystem.

Note that 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 requested bounds. Users who are relying on eigenvalues being contained within the bounds, should test them on return from F02BCF.

If the routine is unable to find a particular vector then on exit this vector will be set to zero.

9 Example

To calculate the eigenvalues whose moduli lie between 4.5 and 6.0, and the corresponding eigenvectors, of the real matrix:

$$\begin{pmatrix} 1.5 & 0.1 & 4.5 & -1.5 \\ -22.5 & 3.5 & 12.5 & -2.5 \\ -2.5 & 0.3 & 4.5 & -2.5 \\ -2.5 & 0.1 & 4.5 & 2.5 \end{pmatrix}.$$

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.

```
*     F02BCF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NMAX, MMAX, IA, IB, IVI, IVR
      PARAMETER        (NMAX=4,MMAX=4,IA=NMAX,IB=NMAX+2,IVI=NMAX,
     +                 IVR=NMAX)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      real             RLB, RUB
      INTEGER          I, IFAIL, J, M, MM, N
*     .. Local Arrays ..
```

```
      real              A(IA,NMAX), B(IB,NMAX), RI(NMAX), RR(NMAX),
     +                  U(NMAX), V(NMAX), VI(IVI,MMAX), VR(IVR,MMAX)
      INTEGER           ICNT(NMAX), INTGER(NMAX)
      LOGICAL           C(NMAX)
*     .. External Subroutines ..
      EXTERNAL          F02BCF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F02BCF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, M
      WRITE (NOUT,*)
      IF (N.LT.1 .OR. N.GT.NMAX .OR. M.LT.1 .OR. M.GT.MMAX) THEN
         WRITE (NOUT,99999) 'N or M out of range: N = ', N, '  M = ', M
         STOP
      END IF
      READ (NIN,*) RLB, RUB, ((A(I,J),J=1,N),I=1,N)
      IFAIL = 1
*
      CALL F02BCF(A,IA,N,RLB,RUB,M,MM,RR,RI,VR,IVR,VI,IVI,INTGER,ICNT,C,
     +            B,IB,U,V,IFAIL)
*
      IF (IFAIL.NE.0) THEN
         WRITE (NOUT,99999) 'Error in F02BCF. IFAIL =', IFAIL
      ELSE
         WRITE (NOUT,*) 'Eigenvalues'
         WRITE (NOUT,99998) (' (',RR(I),',',RI(I),')',I=1,MM)
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Eigenvectors'
         DO 20 I = 1, N
            WRITE (NOUT,99998) (' (',VR(I,J),',',VI(I,J),')',J=1,MM)
   20    CONTINUE
      END IF
      STOP
*
99999 FORMAT (1X,A,I5,A,I5)
99998 FORMAT (1X,4(A,F7.3,A,F7.3,A))
      END
```

9.2 Program Data

```
F02BCF Example Program Data
  4  2
   4.5   6.0
   1.5   0.1   4.5  -1.5
 -22.5   3.5  12.5  -2.5
  -2.5   0.3   4.5  -2.5
  -2.5   0.1   4.5   2.5
```

9.3 Program Results

```
 F02BCF Example Program Results

 Eigenvalues
  (  3.000,  4.000) (  3.000, -4.000)

 Eigenvectors
```

```
( 0.113, -0.151) ( 0.113, 0.151)
( 0.945, 0.000) ( 0.945, 0.000)
( 0.189, 0.000) ( 0.189, 0.000)
( 0.113, -0.151) ( 0.113, 0.151)
```