

## F02BDF – 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

F02BDF calculates selected eigenvalues and eigenvectors of a complex matrix by reduction to Hessenberg form, the *LR* algorithm and inverse iteration, where the moduli of the selected eigenvalues lie between two given values.

### 2 Specification

```

SUBROUTINE F02BDF(AR, IAR, AI, IAI, N, ALB, UB, M, MM, RR, RI, VR,
1                IVR, VI, IVI, INTGER, C, BR, IBR, BI, IBI, U, V,
2                IFAIL)
  INTEGER        IAR, IAI, N, M, MM, IVR, IVI, INTGER(N), IBR,
1                IBI, IFAIL
  real          AR(IAR,N), AI(IAI,N), ALB, UB, RR(N), RI(N),
1                VR(IVR,M), VI(IVI,M), BR(IBR,N), BI(IBM,N),
2                U(N), V(N)
  LOGICAL        C(N)

```

### 3 Description

The complex matrix  $A$  is first reduced to upper Hessenberg form  $H$  using stabilised elementary similarity transformations. All the eigenvalues are then found using the modified *LR* algorithm for complex Hessenberg matrices. Selected eigenvectors of  $H$ , corresponding to the eigenvalues whose moduli lie in the required range, are found using complex 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:** AR(IAR,N) — *real* array *Input/Output*  
*On entry:* the real parts of the elements of the  $n$  by  $n$  complex matrix,  $A$ .  
*On exit:* the real parts of the elements of the derived complex upper Hessenberg matrix  $H$ , together with information about the multipliers used in the reduction.
- 2:** IAR — INTEGER *Input*  
*On entry:* the first dimension of the array AR as declared in the (sub)program from which F02BDF is called.  
*Constraint:*  $IAR \geq N$ .
- 3:** AI(IAI,N) — *real* array *Output*  
*On entry:* the imaginary parts of the elements of the  $n$  by  $n$  complex matrix,  $A$ .  
*On exit:* the imaginary parts of the elements of the derived complex upper Hessenberg matrix, together with information about the multipliers used in the reduction.

- 4:** IAI — INTEGER *Input*  
*On entry:* the first dimension of the array AI as declared in the (sub)program from which F02BDF is called.  
*Constraint:*  $IAI \geq N$ .
- 5:** N — INTEGER *Input*  
*On entry:*  $n$ , the order of the matrix,  $A$ .
- 6:** ALB — *real* *Input*  
**7:** 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.
- 8:** M — INTEGER *Input*  
*On entry:* an upper bound for the number of selected eigenvalues.
- 9:** MM — INTEGER *Output*  
*On exit:* the actual number of selected eigenvalues.
- 10:** RR(N) — *real* array *Output*  
**11:** 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.
- 12:** VR(IVR,M) — *real* array *Output*  
*On exit:* the real parts of the MM selected eigenvectors, stored by columns. The  $i$ th column corresponds to the  $i$ th 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.
- 13:** IVR — INTEGER *Input*  
*On entry:* the first dimension of the array VR as declared in the (sub)program from which F02BDF is called.  
*Constraint:*  $IVR \geq N$ .
- 14:** VI(IVI,M) — *real* array *Output*  
*On exit:* the imaginary parts of the MM selected eigenvectors, stored by columns. The  $i$ th column corresponds to the  $i$ th eigenvalue.
- 15:** IVI — INTEGER *Input*  
*On entry:* the first dimension of the array VI as declared in the (sub)program from which F02BDF is called.  
*Constraint:*  $IVI \geq N$ .
- 16:** 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.
- 17:** C(N) — LOGICAL array *Workspace*  
**18:** BR(IBR,N) — *real* array *Workspace*  
**19:** IBR — INTEGER *Input*  
*On entry:* the first dimension of the array BR as declared in the (sub)program from which F02BDF is called.  
*Constraint:*  $IBR \geq N$ .

**20:** BI(IBM,N) — *real* array

Workspace

**21:** IBI — INTEGER

Input

*On entry:* the first dimension of the array BI as declared in the (sub)program from which F02BDF is called.

*Constraint:*  $IBI \geq N$ .

**22:** U(N) — *real* array

Workspace

**23:** V(N) — *real* array

Workspace

**24:** 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  $30 \times N$  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. Re-run with this value for M.

## 7 Accuracy

The accuracy of the results depends on the conditioning of the original matrix and the multiplicity of the roots. For a detailed error analysis see Wilkinson and Reinsch [1] pp 352, 401 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 then 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 F02BDF.

If the routine is unable to find an eigenvector, then on exit the corresponding locations are set to zero.

## 9 Example

To calculate the eigenvalues whose moduli lie between 5.0 and 7.0, and the corresponding eigenvectors, of the complex matrix:

$$\begin{pmatrix} -21.0-5.0i & 0.0+24.60i & 13.6+10.2i & 0.0+4.0i \\ 0.0+22.5i & 26.00-5.00i & 7.5-10.0i & 2.5+0.0i \\ -2.0+1.5i & 1.68+2.24i & 4.5-5.0i & 1.5+2.0i \\ 0.0-2.5i & -2.60+0.0i & -2.7+3.6i & 2.5-5.0i \end{pmatrix}.$$

## 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.

```

*      F02BDF Example Program Text
*      Mark 14 Revised.  NAG Copyright 1989.
*      .. Parameters ..
      INTEGER          NMAX, MMAX, IAI, IAR, IBI, IBR, IVI, IVR
      PARAMETER        (NMAX=4,MMAX=4,IAI=NMAX,IAR=NMAX,IBI=NMAX,
+                      IBR=NMAX,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             AI(IAI,NMAX), AR(IAR,NMAX), BI(IBI,NMAX),
+                      BR(IBR,NMAX), RI(NMAX), RR(NMAX), U(NMAX),
+                      V(NMAX), VI(IVI,MMAX), VR(IVR,MMAX)
      INTEGER          INTGER(NMAX)
      LOGICAL          C(NMAX)
*      .. External Subroutines ..
      EXTERNAL         F02BDF
*      .. Executable Statements ..
      WRITE (NOUT,*) 'F02BDF 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, ((AR(I,J),AI(I,J),J=1,N),I=1,N)
      IFAIL = 1
*
      CALL F02BDF(AR,IAR,AI,IAI,N,RLB,RUB,M,MM,RR,RI,VR,IVR,VI,IVI,
+              INTGER,C,BR,IBR,BI,IBI,U,V,IFAIL)
*
      IF (IFAIL.NE.0) THEN
         WRITE (NOUT,99999) 'Error in F02BDF. 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

F02BDF Example Program Data

```

4  2
  5.00  7.00
-21.00 -5.00      0 24.60 13.60 10.20      0  4.00
  0.00 22.50 26.00 -5.00  7.50 -10.00  2.50      0
 -2.00  1.50  1.68  2.24  4.50 -5.00  1.50  2.00
      0 -2.50 -2.60      0 -2.70  3.60  2.50 -5.00

```

## 9.3 Program Results

F02BDF Example Program Results

Eigenvalues

```
(  4.000, -5.000) (  2.000, -5.000)
```

Eigenvectors

```

(  0.000,  0.691) (  0.728,  0.000)
(  0.715,  0.000) (  0.000, -0.685)
(  0.005,  0.006) (  0.003, -0.002)
(-0.111,  0.000) (  0.000,  0.043)

```

---