NAG Fortran Library Routine Document

F02FHF

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

F02FHF finds the eigenvalues of the generalized band symmetric eigenvalue problem $Ax = \lambda Bx$, where A and B are symmetric band matrices and B is positive-definite.

2 Specification

```
SUBROUTINE F02FHF(N, MA, A, NRA, MB, B, NRB, D, WORK, LWORK, IFAIL)
INTEGER            N, MA, NRA, MB, NRB, LWORK, IFAIL
real               A(NRA,N), B(NRB,N), D(N), WORK(LWORK)
```

3 Description

The generalized band symmetric eigenvalue problem $Ax = \lambda Bx$, where A is a symmetric band matrix of band width $2m_A + 1$ and B is a positive-definite symmetric band matrix of band width $2m_B + 1$, is solved by a variant of the method of Crawford.

The routine first transforms the problem $Ax = \lambda Bx$ to a standard band symmetric eigenvalue problem $Cy = \lambda y$, where C is a band symmetric matrix of band width $2m_A + 1$, using F01BUF and F01BVF. This step involves the implicit inversion of the matrix B and so this routine should be used with caution if B is ill-conditioned with respect to inversion.

The eigenvalues of the standard problem $Cy = \lambda y$ are then obtained by reducing C to tridiagonal form and then applying the QL variant of the QR algorithm to the tridiagonal form, using F08HEF (SSBTRD/DSBTRD) and F08JFF (SSTERF/DSTERF). The above-mentioned routines should be consulted for further information on the methods used.

Once the eigenvalues have been found by this routine, selected eigenvectors may be obtained by repeated calls to F02SDF with the original matrices A and B as data.

The routine assumes that $m_A \geq m_B$ and hence if the band width of A is actually smaller than that of B, then A must be filled out with additional zero diagonals.

4 References

Crawford C R (1973) Reduction of a band-symmetric generalized eigenvalue problem *Comm. ACM* **16** 41–44

Wilkinson J H (1977) Some recent advances in numerical linear algebra *The State of the Art in Numerical Analysis* (ed D A H Jacobs) Academic Press

5 Parameters

1: N – INTEGER *Input*

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

Constraint: N $\geq$ 1.

2: MA – INTEGER *Input*

On entry: m_A, the number of super-diagonals within the band of A. Normally $m_A \ll n$.

Constraint: $0 \leq$ MA $\leq$ N $-$ 1.

3: A(NRA,N) – ***real*** array *Input/Output*

On entry: the upper triangle of the n by n symmetric band matrix A, with the diagonal of the matrix stored in the $(m_A + 1)$th row of the array, and the m_A super-diagonals within the band stored in the first m_A rows of the array. Each column of the matrix is stored in the corresponding column of the array. For example, if $n = 6$ and $m = 2$, the storage space is

$$\begin{matrix} * & * & a_{13} & a_{24} & a_{35} & a_{46} \\ * & a_{12} & a_{23} & a_{34} & a_{45} & a_{56} \\ a_{11} & a_{22} & a_{33} & a_{44} & a_{55} & a_{66} \end{matrix}$$

Elements in the top left corner of the array need not be set. The following code assigns the matrix elements within the band to the correct elements of the array:

```
      MA1 = MA + 1
      DO 20 J = 1, N
         DO 10 I = MAX(1,J-MA1+1), J
            A(I-J+MA1,J) = matrix (I,J)
   10    CONTINUE
   20 CONTINUE
```

On exit: A is overwritten by the corresponding elements of C.

4: NRA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F02FHF is called.

Constraint: NRA $\geq$ MA $+ 1$.

5: MB – INTEGER *Input*

On entry: m_B, the number of super-diagonals within the band of B.

Constraint: $0 \leq$ MB $\leq$ MA.

6: B(NRB,N) – ***real*** array *Input/Output*

On entry: the upper triangle of the n by n symmetric positive-definite band matrix B, with the diagonal of the matrix stored in the $(m_B + 1)$th row of the array, and the m_B super-diagonals within the band stored in the first m_B rows of the array. Each column of the matrix is stored in the corresponding column of the array.

On exit: B is overwritten.

7: NRB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F02FHF is called.

Constraint: NRB $\geq$ MB $+ 1$.

8: D(N) – ***real*** array *Output*

On exit: the eigenvalues in descending order of magnitude.

9: WORK(LWORK) – ***real*** array *Workspace*
10: LWORK – INTEGER *Input*

On entry: the length of the array WORK, as declared in the (sub)program from which F02FHF is called.

Constraint: LWORK $\geq$ max(N, (3 $\times$ MA $+$ MB) $\times$ (MA $+$ MB $+ 1$)).

11: 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, $\mathrm{N} < 1$,
or $\mathrm{MA} < 0$,
or $\mathrm{MA} \geq \mathrm{N}$,
or $\mathrm{MB} < 0$,
or $\mathrm{MB} > \mathrm{MA}$,
or $\mathrm{NRA} \leq \mathrm{MA}$,
or $\mathrm{NRB} \leq \mathrm{MB}$,
or $\mathrm{LWORK} < \max(\mathrm{N}, (3 \times \mathrm{MA} + \mathrm{MB}) \times (\mathrm{MA} + \mathrm{MB} + 1))$.

IFAIL $= 2$

The matrix B is either not positive-definite or is nearly singular.

IFAIL $= 3$

This failure is very unlikely to occur, but indicates that more than $30 \times \mathrm{N}$ iterations are required by the QR part of the algorithm. The input parameters should be carefully checked to ensure that the error is not due to an incorrect parameter.

7 Accuracy

The computed eigenvalues will be the exact eigenvalues of a neighbouring problem $(A+E)x = \lambda(B+F)x$, where $\|E\|$ and $\|F\|$ are of the order of $\epsilon c(B)\|A\|$ and $\epsilon c(B)\|B\|$ respectively, where $c(B)$ is the condition number of B with respect to inversion and ϵ is the ***machine precision***.

Thus if B is ill-conditioned with respect to inversion there may be a severe loss of accuracy in well-conditioned eigenvalues.

8 Further Comments

The time taken by the routine is very approximately proportional to $n^2\left(\frac{m_A + m_B + 2}{m_A} + \frac{m_B^2}{8}\right)$, provided $m_A > 0$.

9 Example

To find the eigenvalues of the generalized band symmetric eigenvalue problem $Ax = \lambda Bx$, where

$$A = \begin{pmatrix} 5 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 6 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & 7 & 3 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 3 & 8 & 4 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 4 & 9 & 4 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 4 & 8 & 3 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 3 & 7 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & 6 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 5 \end{pmatrix}$$

and

$$B = \begin{pmatrix} 4 & 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 5 & 1 & -2 & 0 & 0 & 0 & 0 & 0 \\ -2 & 1 & 6 & 1 & -2 & 0 & 0 & 0 & 0 \\ 0 & -2 & 1 & 6 & 1 & -2 & 0 & 0 & 0 \\ 0 & 0 & -2 & 1 & 6 & 1 & -2 & 0 & 0 \\ 0 & 0 & 0 & -2 & 1 & 6 & 1 & -2 & 0 \\ 0 & 0 & 0 & 0 & -2 & 1 & 6 & 1 & -2 \\ 0 & 0 & 0 & 0 & 0 & -2 & 1 & 6 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2 & 1 & 6 \end{pmatrix}.$$

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.

```
*     F02FHF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NMAX, MAMAX, MBMAX, NRA, NRB, LWORK
      PARAMETER        (NMAX=20,MAMAX=5,MBMAX=5,NRA=MAMAX+1,NRB=MBMAX+1,
     +                 LWORK=NMAX+(3*MAMAX+MBMAX)*(MAMAX+MBMAX+2))
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, J, MA, MB, N
*     .. Local Arrays ..
      real             A(NRA,NMAX), B(NRB,NMAX), D(NMAX), WORK(LWORK)
*     .. External Subroutines ..
      EXTERNAL         F02FHF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F02FHF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, MA, MB
      WRITE (NOUT,*)
      IF (N.LT.1 .OR. N.GT.NMAX .OR. MA.LT.0 .OR. MA.GT.MAMAX .OR.
     +    MB.LT.0 .OR. MB.GT.MBMAX) THEN
         WRITE (NOUT,*) 'N or MA or MB is out of range.'
         WRITE (NOUT,99999) 'N = ', N, '   MA = ', MA, '   MB = ', MB
      ELSE
         DO 20 I = 1, MA + 1
            READ (NIN,*) (A(I,J),J=1,N)
   20    CONTINUE
         DO 40 I = 1, MB + 1
            READ (NIN,*) (B(I,J),J=1,N)
   40    CONTINUE
*
         IFAIL = 1
*
```

```
         CALL F02FHF(N,MA,A,NRA,MB,B,NRB,D,WORK,LWORK,IFAIL)
*
         IF (IFAIL.NE.0) THEN
            WRITE (NOUT,*)
            WRITE (NOUT,99999) 'F02FHF fails. IFAIL =', IFAIL
         ELSE
            WRITE (NOUT,*) 'Eigenvalues'
            WRITE (NOUT,99998) (D(J),J=1,N)
         END IF
      END IF
      STOP
*
99999 FORMAT (1X,A,I5,A,I5,A,I5)
99998 FORMAT (1X,7F9.4)
      END
```

9.2 Program Data

```
F02FHF Example Program Data
   9   2   2
  0.0   0.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0  -1.0
  0.0   1.0   2.0   3.0   4.0   4.0   3.0   2.0   1.0
  5.0   6.0   7.0   8.0   9.0   8.0   7.0   6.0   5.0
  0.0   0.0  -2.0  -2.0  -2.0  -2.0  -2.0  -2.0  -2.0
  0.0   2.0   1.0   1.0   1.0   1.0   1.0   1.0   1.0
  4.0   5.0   6.0   6.0   6.0   6.0   6.0   6.0   6.0
```

9.3 Program Results

```
 F02FHF Example Program Results

 Eigenvalues
    0.0544   0.7578   0.8277   0.9188   0.9429   1.1667   1.5582
    2.6623   4.7791
```