NAG Library Routine Document

F07VHF (DTBRFS)

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

F07VHF (DTBRFS) returns error bounds for the solution of a real triangular band system of linear equations with multiple right-hand sides, $AX = B$ or $A^{\mathrm{T}}X = B$.

2 Specification

```
 SUBROUTINE F07VHF(UPLO, TRANS, DIAG, N, KD, NRHS, AB, LDAB, B, LDB, X,
1                  LDX, FERR, BERR, WORK, IWORK, INFO)
 INTEGER           N, KD, NRHS, LDAB, LDB, LDX, IWORK(N), INFO
 double precision  AB(LDAB,*), B(LDB,*), X(LDX,*), FERR(NRHS),
1                  BERR(NRHS), WORK(3*N)
 CHARACTER*1       UPLO, TRANS, DIAG
```

The routine may be called by its LAPACK name ***dtbrfs***.

3 Description

F07VHF (DTBRFS) returns the backward errors and estimated bounds on the forward errors for the solution of a real triangular band system of linear equations with multiple right-hand sides $AX = B$ or $A^{\mathrm{T}}X = B$. The routine handles each right-hand side vector (stored as a column of the matrix B) independently, so we describe the function of F07VHF (DTBRFS) in terms of a single right-hand side b and solution x.

Given a computed solution x, the routine computes the *component-wise backward error* β. This is the size of the smallest relative perturbation in each element of A and b such that x is the exact solution of a perturbed system

$$(A + \delta A)x = b + \delta b$$
$$|\delta a_{ij}| \leq \beta |a_{ij}| \quad \text{and} \quad |\delta b_i| \leq \beta |b_i|.$$

Then the routine estimates a bound for the *component-wise forward error* in the computed solution, defined by:

$$\max_i |x_i - \hat{x}_i| / \max_i |x_i|$$

where $\hat{x}$ is the true solution.

For details of the method, see the F07 Chapter Introduction.

4 References

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

5 Parameters

1: UPLO – CHARACTER*1 *Input*

On entry: indicates whether A is upper or lower triangular.

UPLO = 'U'
A is upper triangular.

UPLO = 'L'
A is lower triangular.

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

2: TRANS – CHARACTER*1 *Input*

On entry: indicates the form of the equations.

TRANS = 'N'
The equations are of the form $AX = B$.

TRANS = 'T' or 'C'
The equations are of the form $A^{\mathrm{T}}X = B$.

Constraint: TRANS = 'N', 'T' or 'C'.

3: DIAG – CHARACTER*1 *Input*

On entry: indicates whether A is a nonunit or unit triangular matrix.

DIAG = 'N'
A is a nonunit triangular matrix.

DIAG = 'U'
A is a unit triangular matrix; the diagonal elements are not referenced and are assumed to be 1.

Constraint: DIAG = 'N' or 'U'.

4: N – INTEGER *Input*

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

Constraint: N ≥ 0.

5: KD – INTEGER *Input*

On entry: k_d, the number of superdiagonals of the matrix A if UPLO = 'U', or the number of subdiagonals if UPLO = 'L'.

Constraint: KD ≥ 0.

6: NRHS – INTEGER *Input*

On entry: r, the number of right-hand sides.

Constraint: NRHS ≥ 0.

7: AB(LDAB,$*$) – ***double precision*** array *Input*

Note: the second dimension of the array AB must be at least $\max(1, \mathrm{N})$.

On entry: the n by n triangular band matrix A.

The matrix is stored in rows 1 to $k_d + 1$, more precisely,

if UPLO = 'U', the elements of the upper triangle of A within the band must be stored with element A_{ij} in $\mathrm{AB}(k_d + 1 + i - j, j)$ for $\max(1, j - k_d) \leq i \leq j$;

if UPLO = 'L', the elements of the lower triangle of A within the band must be stored with element A_{ij} in $\mathrm{AB}(1 + i - j, j)$ for $j \leq i \leq \min(n, j + k_d)$.

If DIAG = 'U', the diagonal elements of A are assumed to be 1, and are not referenced.

8: LDAB – INTEGER *Input*

On entry: the first dimension of the array AB as declared in the (sub)program from which F07VHF (DTBRFS) is called.

Constraint: LDAB $\geq$ KD + 1.

9: B(LDB,$*$) – ***double precision*** array *Input*

Note: the second dimension of the array B must be at least $\max(1, \text{NRHS})$.

On entry: the n by r right-hand side matrix B.

10: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F07VHF (DTBRFS) is called.

Constraint: LDB $\geq \max(1, \text{N})$.

11: X(LDX,$*$) – ***double precision*** array *Input*

Note: the second dimension of the array X must be at least $\max(1, \text{NRHS})$.

On entry: the n by r solution matrix X, as returned by F07VEF (DTBTRS).

12: LDX – INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which F07VHF (DTBRFS) is called.

Constraint: LDX $\geq \max(1, \text{N})$.

13: FERR(NRHS) – ***double precision*** array *Output*

On exit: FERR(j) contains an estimated error bound for the jth solution vector, that is, the jth column of X, for $j = 1, 2, \ldots, r$.

14: BERR(NRHS) – ***double precision*** array *Output*

On exit: BERR(j) contains the component-wise backward error bound β for the jth solution vector, that is, the jth column of X, for $j = 1, 2, \ldots, r$.

15: WORK($3 \times$ N) – ***double precision*** array *Workspace*

16: IWORK(N) – INTEGER array *Workspace*

17: INFO – INTEGER *Output*

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

6 Error Indicators and Warnings

Errors or warnings detected by the routine:

INFO < 0

If INFO $= -i$, the ith parameter had an illegal value. An explanatory message is output, and execution of the program is terminated.

7 Accuracy

The bounds returned in FERR are not rigorous, because they are estimated, not computed exactly; but in practice they almost always overestimate the actual error.

8 Further Comments

A call to F07VHF (DTBRFS), for each right-hand side, involves solving a number of systems of linear equations of the form $Ax = b$ or $A^{\mathrm{T}}x = b$; the number is usually 4 or 5 and never more than 11. Each solution involves approximately $2nk$ floating-point operations (assuming $n \gg k$).

The complex analogue of this routine is F07VVF (ZTBRFS).

9 Example

This example solves the system of equations $AX = B$ and to compute forward and backward error bounds, where

$$A = \begin{pmatrix} -4.16 & 0.00 & 0.00 & 0.00 \\ -2.25 & 4.78 & 0.00 & 0.00 \\ 0.00 & 5.86 & 6.32 & 0.00 \\ 0.00 & 0.00 & -4.82 & 0.16 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} -16.64 & -4.16 \\ -13.78 & -16.59 \\ 13.10 & -4.94 \\ -14.14 & -9.96 \end{pmatrix}.$$

9.1 Program Text

```
*     F07VHF Example Program Text
*     Mark 15 Release. NAG Copyright 1991.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, KDMAX, LDAB, NRHMAX, LDB, LDX
      PARAMETER        (NMAX=8,KDMAX=NMAX,LDAB=KDMAX+1,NRHMAX=NMAX,
     +                 LDB=NMAX,LDX=NMAX)
      CHARACTER        TRANS, DIAG
      PARAMETER        (TRANS='N',DIAG='N')
*     .. Local Scalars ..
      INTEGER          I, IFAIL, INFO, J, KD, N, NRHS
      CHARACTER        UPLO
*     .. Local Arrays ..
      DOUBLE PRECISION AB(LDAB,NMAX), B(LDB,NRHMAX), BERR(NRHMAX),
     +                 FERR(NRHMAX), WORK(3*NMAX), X(LDX,NMAX)
      INTEGER          IWORK(NMAX)
*     .. External Subroutines ..
      EXTERNAL         DTBRFS, DTBTRS, F06QFF, X04CAF
*     .. Intrinsic Functions ..
      INTRINSIC        MAX, MIN
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F07VHF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, KD, NRHS
      IF (N.LE.NMAX .AND. KD.LE.KDMAX .AND. NRHS.LE.NRHMAX) THEN
*
*        Read A and B from data file, and copy B to X
*
         READ (NIN,*) UPLO
         IF (UPLO.EQ.'U') THEN
            DO 20 I = 1, N
               READ (NIN,*) (AB(KD+1+I-J,J),J=I,MIN(N,I+KD))
   20       CONTINUE
         ELSE IF (UPLO.EQ.'L') THEN
            DO 40 I = 1, N
               READ (NIN,*) (AB(1+I-J,J),J=MAX(1,I-KD),I)
   40       CONTINUE
         END IF
         READ (NIN,*) ((B(I,J),J=1,NRHS),I=1,N)
*
         CALL F06QFF('General',N,NRHS,B,LDB,X,LDX)
*
*        Compute solution in the array X
*
         CALL DTBTRS(UPLO,TRANS,DIAG,N,KD,NRHS,AB,LDAB,X,LDX,INFO)
*
```

```
*        Compute backward errors and estimated bounds on the
*        forward errors
*
         CALL DTBRFS(UPLO,TRANS,DIAG,N,KD,NRHS,AB,LDAB,B,LDB,X,LDX,FERR,
     +              BERR,WORK,IWORK,INFO)
*
*        Print solution
*
         WRITE (NOUT,*)
         IFAIL = 0
         CALL X04CAF('General',' ',N,NRHS,X,LDX,'Solution(s)',IFAIL)
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Backward errors (machine-dependent)'
         WRITE (NOUT,99999) (BERR(J),J=1,NRHS)
         WRITE (NOUT,*)
     +     'Estimated forward error bounds (machine-dependent)'
         WRITE (NOUT,99999) (FERR(J),J=1,NRHS)
      END IF
*
99999 FORMAT ((3X,1P,7E11.1))
      END
```

9.2 Program Data

```
F07VHF Example Program Data
  4  1  2                    :Values of N, KD and NRHS
  'L'                         :Value of UPLO
 -4.16
 -2.25   4.78
         5.86   6.32
               -4.82   0.16   :End of matrix A
-16.64  -4.16
-13.78 -16.59
 13.10  -4.94
-14.14  -9.96                 :End of matrix B
```

9.3 Program Results

```
 F07VHF Example Program Results

 Solution(s)
             1          2
 1      4.0000     1.0000
 2     -1.0000    -3.0000
 3      3.0000     2.0000
 4      2.0000    -2.0000

 Backward errors (machine-dependent)
       4.7E-17    2.5E-17
 Estimated forward error bounds (machine-dependent)
       5.4E-14    5.8E-14
```