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,          &
                   LDX, FERR, BERR, WORK, IWORK, INFO)

INTEGER            N, KD, NRHS, LDAB, LDB, LDX, IWORK(N), INFO
REAL (KIND=nag_wp) AB(LDAB,*), B(LDB,*), X(LDX,*), FERR(NRHS), 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

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

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: specifies 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,$*$) – REAL (KIND=nag_wp) 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 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 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,$*$) – REAL (KIND=nag_wp) 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,$*$) – REAL (KIND=nag_wp) 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) – REAL (KIND=nag_wp) 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) – REAL (KIND=nag_wp) 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) – REAL (KIND=nag_wp) 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

```
    Program f07vhfe

!     F07VHF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: dtbrfs, dtbtrs, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter                :: nin = 5, nout = 6
      Character (1), Parameter          :: diag = 'N', trans = 'N'
!     .. Local Scalars ..
      Integer                           :: i, ifail, info, j, kd, ldab, ldb,    &
                                           ldx, n, nrhs
      Character (1)                     :: uplo
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable   :: ab(:,:), b(:,:), berr(:), ferr(:),   &
                                           work(:), x(:,:)
      Integer, Allocatable              :: iwork(:)
!     .. Intrinsic Procedures ..
      Intrinsic                         :: max, min
!     .. Executable Statements ..
      Write (nout,*) 'F07VHF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n, kd, nrhs
      ldab = kd + 1
      ldb = n
```

```
      ldx = n
      Allocate (ab(ldab,n),b(ldb,nrhs),berr(nrhs),ferr(nrhs),work(3*n), &
        x(ldx,n),iwork(n))

!     Read A and B from data file, and copy B to X

      Read (nin,*) uplo
      If (uplo=='U') Then
        Do i = 1, n
          Read (nin,*)(ab(kd+1+i-j,j),j=i,min(n,i+kd))
        End Do
      Else If (uplo=='L') Then
        Do i = 1, n
          Read (nin,*)(ab(1+i-j,j),j=max(1,i-kd),i)
        End Do
      End If
      Read (nin,*)(b(i,1:nrhs),i=1,n)

      x(1:n,1:nrhs) = b(1:n,1:nrhs)

!     Compute solution in the array X
!     The NAG name equivalent of dtbtrs is f07vef
      Call dtbtrs(uplo,trans,diag,n,kd,nrhs,ab,ldab,x,ldx,info)

!     Compute backward errors and estimated bounds on the
!     forward errors

!     The NAG name equivalent of dtbrfs is f07vhf
      Call dtbrfs(uplo,trans,diag,n,kd,nrhs,ab,ldab,b,ldb,x,ldx,ferr,berr, &
        work,iwork,info)

!     Print solution

      Write (nout,*)
      Flush (nout)

!     ifail: behaviour on error exit
!             =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 0
      Call x04caf('General',' ',n,nrhs,x,ldx,'Solution(s)',ifail)

      Write (nout,*)
      Write (nout,*) 'Backward errors (machine-dependent)'
      Write (nout,99999) berr(1:nrhs)
      Write (nout,*) 'Estimated forward error bounds (machine-dependent)'
      Write (nout,99999) ferr(1:nrhs)

99999 Format ((3X,1P,7E11.1))
    End Program f07vhfe
```

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
```