NAG Library Routine Document

F07UHF (DTPRFS)

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

F07UHF (DTPRFS) returns error bounds for the solution of a real triangular system of linear equations with multiple right-hand sides, $AX = B$ or $A^{\mathrm{T}}X = B$, using packed storage.

2 Specification

```
SUBROUTINE F07UHF (UPLO, TRANS, DIAG, N, NRHS, AP, B, LDB, X, LDX, FERR,        &
                   BERR, WORK, IWORK, INFO)

INTEGER              N, NRHS, LDB, LDX, IWORK(N), INFO
REAL (KIND=nag_wp) AP(*), 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 ***dtprfs***.

3 Description

F07UHF (DTPRFS) returns the backward errors and estimated bounds on the forward errors for the solution of a real triangular system of linear equations with multiple right-hand sides $AX = B$ or $A^{\mathrm{T}}X = B$, using packed storage. The routine handles each right-hand side vector (stored as a column of the matrix B) independently, so we describe the function of F07UHF (DTPRFS) 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

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

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

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

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: $\mathrm{N} \geq 0$.

5: NRHS – INTEGER *Input*

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

Constraint: $\mathrm{NRHS} \geq 0$.

6: AP($*$) – REAL (KIND=nag_wp) array *Input*

Note: the dimension of the array AP must be at least $\max(1, \mathrm{N} \times (\mathrm{N}+1)/2)$.

On entry: the n by n triangular matrix A, packed by columns.

More precisely,

if UPLO = 'U', the upper triangle of A must be stored with element A_{ij} in $\mathrm{AP}(i + j(j-1)/2)$ for $i \leq j$;

if UPLO = 'L', the lower triangle of A must be stored with element A_{ij} in $\mathrm{AP}(i + (2n-j)(j-1)/2)$ for $i \geq j$.

If DIAG = 'U', the diagonal elements of A are assumed to be 1, and are not referenced; the same storage scheme is used whether DIAG = 'N' or ‘U’.

7: B(LDB,$*$) – REAL (KIND=nag_wp) array *Input*

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

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

8: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F07UHF (DTPRFS) is called.

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

9: X(LDX,$*$) – REAL (KIND=nag_wp) array *Input*

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

On entry: the n by r solution matrix X, as returned by F07UEF (DTPTRS).

10: LDX – INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which F07UHF (DTPRFS) is called.

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

11: FERR(NRHS) – REAL (KIND=nag_wp) array *Output*

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

12: BERR(NRHS) – REAL (KIND=nag_wp) array *Output*

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

13: WORK($3 \times \mathrm{N}$) – REAL (KIND=nag_wp) array *Workspace*

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

15: INFO – INTEGER *Output*

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

6 Error Indicators and Warnings

Errors or warnings detected by the routine:

$\mathrm{INFO} < 0$

If $\mathrm{INFO} = -i$, 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 F07UHF (DTPRFS), 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 n^2 floating point operations.

The complex analogue of this routine is F07UVF (ZTPRFS).

9 Example

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

$$A = \begin{pmatrix} 4.30 & 0.00 & 0.00 & 0.00 \\ -3.96 & -4.87 & 0.00 & 0.00 \\ 0.40 & 0.31 & -8.02 & 0.00 \\ -0.27 & 0.07 & -5.95 & 0.12 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} -12.90 & -21.50 \\ 16.75 & 14.93 \\ -17.55 & 6.33 \\ -11.04 & 8.09 \end{pmatrix},$$

using packed storage for A.

9.1 Program Text

```
    Program f07uhfe

!     F07UHF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: dtprfs, dtptrs, 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, ldb, ldx, n, nrhs
      Character (1)                    :: uplo
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: ap(:), b(:,:), berr(:), ferr(:),     &
                                          work(:), x(:,:)
      Integer, Allocatable             :: iwork(:)
!     .. Executable Statements ..
      Write (nout,*) 'F07UHF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n, nrhs
      ldb = n
      ldx = n
      Allocate (ap(n*(n+1)/2),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
        Read (nin,*)((ap(i+j*(j-1)/2),j=i,n),i=1,n)
      Else If (uplo=='L') Then
        Read (nin,*)((ap(i+(2*n-j)*(j-1)/2),j=1,i),i=1,n)
      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 dtptrs is f07uef
      Call dtptrs(uplo,trans,diag,n,nrhs,ap,x,ldx,info)

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

!     The NAG name equivalent of dtprfs is f07uhf
      Call dtprfs(uplo,trans,diag,n,nrhs,ap,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 f07uhfe
```

9.2 Program Data

```
F07UHF Example Program Data
  4  2                         :Values of N and NRHS
  'L'                          :Value of UPLO
  4.30
 -3.96  -4.87
  0.40   0.31  -8.02
 -0.27   0.07  -5.95   0.12   :End of matrix A
-12.90 -21.50
 16.75  14.93
-17.55   6.33
-11.04   8.09                  :End of matrix B
```

9.3 Program Results

```
 F07UHF Example Program Results

 Solution(s)
             1          2
 1     -3.0000    -5.0000
 2     -1.0000     1.0000
 3      2.0000    -1.0000
 4      1.0000     6.0000

 Backward errors (machine-dependent)
       6.9E-17    0.0E+00
 Estimated forward error bounds (machine-dependent)
       8.3E-14    2.6E-14
```