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,
1                  BERR, WORK, IWORK, INFO)
 INTEGER           N, NRHS, LDB, LDX, IWORK(N), INFO
 double precision  AP(*), B(LDB,*), X(LDX,*), FERR(NRHS), BERR(NRHS),
1                  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

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

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: NRHS – INTEGER *Input*

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

Constraint: NRHS ≥ 0.

6: AP$(*)$ – ***double precision*** 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 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 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,$*$) – ***double precision*** 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: LDB $\geq \max(1, \mathrm{N})$.

9: X(LDX,$*$) – ***double precision*** 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: LDX $\geq \max(1, \mathrm{N})$.

11: 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 $j = 1, 2, \ldots, r$.

12: 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 $j = 1, 2, \ldots, r$.

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

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

15: 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 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

```
*     F07UHF Example Program Text
*     Mark 15 Release. NAG Copyright 1991.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, NRHMAX, LDB, LDX
      PARAMETER        (NMAX=8,NRHMAX=NMAX,LDB=NMAX,LDX=NMAX)
      CHARACTER        TRANS, DIAG
      PARAMETER        (TRANS='N',DIAG='N')
*     .. Local Scalars ..
      INTEGER          I, IFAIL, INFO, J, N, NRHS
      CHARACTER        UPLO
*     .. Local Arrays ..
      DOUBLE PRECISION AP(NMAX*(NMAX+1)/2), B(LDB,NRHMAX), BERR(NRHMAX),
     +                 FERR(NRHMAX), WORK(3*NMAX), X(LDX,NMAX)
      INTEGER          IWORK(NMAX)
*     .. External Subroutines ..
      EXTERNAL         DTPRFS, DTPTRS, F06QFF, X04CAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F07UHF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, NRHS
      IF (N.LE.NMAX .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
            READ (NIN,*) ((AP(I+J*(J-1)/2),J=I,N),I=1,N)
         ELSE IF (UPLO.EQ.'L') THEN
            READ (NIN,*) ((AP(I+(2*N-J)*(J-1)/2),J=1,I),I=1,N)
         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 DTPTRS(UPLO,TRANS,DIAG,N,NRHS,AP,X,LDX,INFO)
*
*        Compute backward errors and estimated bounds on the
*        forward errors
*
         CALL DTPRFS(UPLO,TRANS,DIAG,N,NRHS,AP,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

```
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
```