

F07THF (STRRFS/DTRRFS) – NAG Fortran Library Routine Document

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

F07THF (STRRFS/DTRRFS) returns error bounds for the solution of a real triangular system of linear equations with multiple right-hand sides, $AX = B$ or $A^T X = B$.

2 Specification

```

SUBROUTINE F07THF(UPLO, TRANS, DIAG, N, NRHS, A, LDA, B, LDB, X,
1          LDX, FERR, BERR, WORK, IWORK, INFO)
ENTRY      strrfs(UPLO, TRANS, DIAG, N, NRHS, A, LDA, B, LDB, X,
1          LDX, FERR, BERR, WORK, IWORK, INFO)
INTEGER    N, NRHS, LDA, LDB, LDX, IWORK(*), INFO
real      A(LDA,*), B(LDB,*), X(LDX,*), FERR(*), BERR(*),
1          WORK(*)
CHARACTER*1 UPLO, TRANS, DIAG

```

The ENTRY statement enables the routine to be called by its LAPACK name.

3 Description

This routine 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 $A^T X = B$. The routine handles each right-hand side vector (stored as a column of the matrix B) independently, so we describe the function of the routine 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 Chapter Introduction.

4 References

- [1] Golub G H and van Loan C F (1996) *Matrix Computations* Johns Hopkins University Press (3rd Edition), Baltimore

5 Parameters

1: UPLO — CHARACTER*1

Input

On entry: indicates whether A is upper or lower triangular as follows:

- if UPLO = 'U', then A is upper triangular;
- if UPLO = 'L', then A is lower triangular.

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

2: TRANS — CHARACTER*1 *Input*

On entry: indicates the form of the equations as follows:

- if TRANS = 'N', then the equations are of the form $AX = B$;
- if TRANS = 'T' or 'C', then the equations are of the form $A^T X = B$.

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

3: DIAG — CHARACTER*1 *Input*

On entry: indicates whether A is a non-unit or unit triangular matrix as follows:

- if DIAG = 'N', then A is a non-unit triangular matrix;
- if DIAG = 'U', then 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: $N \geq 0$.

5: NRHS — INTEGER *Input*

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

Constraint: NRHS ≥ 0 .

6: A(LDA,*) — *real* array *Input*

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

On entry: the n by n triangular matrix A . If UPLO = 'U', A is upper triangular and the elements of the array below the diagonal are not referenced; if UPLO = 'L', A is lower triangular and the elements of the array above the diagonal are not referenced. If DIAG = 'U', the diagonal elements of A are not referenced, but are assumed to be 1.

7: LDA — INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F07THF (STRRFS/DTRRFS) is called.

Constraint: LDA $\geq \max(1, N)$.

8: B(LDB,*) — *real* 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 .

9: LDB — INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F07THF (STRRFS/DTRRFS) is called.

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

10: X(LDX,*) — *real* 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 F07TEF (STRTRS/DTRTRS).

- 11:** LDX — INTEGER *Input*
On entry: the first dimension of the array X as declared in the (sub)program from which F07THF (STRRFS/DTRRFS) is called.
Constraint: $LDX \geq \max(1, N)$.
- 12:** FERR(*) — *real* array *Output*
Note: the dimension of the array FERR must be at least $\max(1, NRHS)$.
On exit: FERR(*j*) contains an estimated error bound for the *j*th solution vector, that is, the *j*th column of *X*, for $j = 1, 2, \dots, r$.
- 13:** BERR(*) — *real* array *Output*
Note: the dimension of the array BERR must be at least $\max(1, NRHS)$.
On exit: BERR(*j*) contains the component-wise backward error bound β for the *j*th solution vector, that is, the *j*th column of *X*, for $j = 1, 2, \dots, r$.
- 14:** WORK(*) — *real* array *Workspace*
Note: the dimension of the array WORK must be at least $\max(1, 3*N)$.
- 15:** IWORK(*) — INTEGER array *Workspace*
Note: the dimension of the array IWORK must be at least $\max(1, N)$.
- 16:** INFO — INTEGER *Output*
On exit: INFO = 0 unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

INFO < 0

If INFO = $-i$, the *i*th 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 this routine involves, for each right-hand side, solving a number of systems of linear equations of the form $Ax = b$ or $A^T x = b$; 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 F07TVF (CTRRFS/ZTRRFS).

9 Example

To solve the system of equations $AX = B$ and to compute forward and backward error bounds, where

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

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.

```

*   F07THF Example Program Text
*   Mark 15 Release. NAG Copyright 1991.
*   .. Parameters ..
INTEGER          NIN, NOUT
PARAMETER       (NIN=5,NOUT=6)
INTEGER          NMAX, NRHMAX, LDA, LDB, LDX
PARAMETER       (NMAX=8,NRHMAX=NMAX,LDA=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 ..
real           A(LDA,NMAX), B(LDB,NRHMAX), BERR(NRHMAX),
+               FERR(NRHMAX), WORK(3*NMAX), X(LDX,NMAX)
INTEGER          IWORK(NMAX)
*   .. External Subroutines ..
EXTERNAL         strrfs, strtrs, F06QFF, X04CAF
*   .. Executable Statements ..
WRITE (NOUT,*) 'F07THF 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,*) ((A(I,J),J=I,N),I=1,N)
    ELSE IF (UPLO.EQ.'L') THEN
        READ (NIN,*) ((A(I,J),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 strtrs(UPLO,TRANS,DIAG,N,NRHS,A,LDA,X,LDX,INFO)
*
*       Compute backward errors and estimated bounds on the
*       forward errors
*
    CALL strrfs(UPLO,TRANS,DIAG,N,NRHS,A,LDA,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
    STOP
*
99999 FORMAT ((3X,1P,7e11.1))
    END

```

9.2 Program Data

F07THF 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

F07THF 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
---------	---------