

NAG Library Routine Document

F07TVF (ZTRRFS)

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

F07TVF (ZTRRFS) returns error bounds for the solution of a complex triangular system of linear equations with multiple right-hand sides, $AX = B$, $A^T X = B$ or $A^H X = B$.

2 Specification

```

SUBROUTINE F07TVF(UPLO, TRANS, DIAG, N, NRHS, A, LDA, B, LDB, X, LDX,
1              FERR, BERR, WORK, RWORK, INFO)
INTEGER          N, NRHS, LDA, LDB, LDX, INFO
double precision FERR(NRHS), BERR(NRHS), RWORK(N)
complex*16      A(LDA,*), B(LDB,*), X(LDX,*), WORK(2*N)
CHARACTER*1      UPLO, TRANS, DIAG

```

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

3 Description

F07TVF (ZTRRFS) returns the backward errors and estimated bounds on the forward errors for the solution of a complex triangular system of linear equations with multiple right-hand sides $AX = B$, $A^T X = B$ or $A^H X = B$. The routine handles each right-hand side vector (stored as a column of the matrix B) independently, so we describe the function of F07TVF (ZTRRFS) 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

$$| \delta a_{ij} | \leq \beta | a_{ij} | \quad \text{and} \quad | \delta b_i | \leq \beta | b_i |.$$

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'
 The equations are of the form $A^T X = B$.
 TRANS = 'C'
 The equations are of the form $A^H X = B$.
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: $N \geq 0$.
- 5: NRHS – INTEGER *Input*
On entry: r , the number of right-hand sides.
Constraint: NRHS ≥ 0 .
- 6: A(LDA,*) – **complex*16** 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 assumed to be 1, and are not referenced.
- 7: LDA – INTEGER *Input*
On entry: the first dimension of the array A as declared in the (sub)program from which F07TVF (ZTRRFS) is called.
Constraint: LDA $\geq \max(1, N)$.

- 8: B(LDB,*) – **complex*16** 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 F07TVF (ZTRRFS) is called.
Constraint: $\text{LDB} \geq \max(1, N)$.
- 10: X(LDX,*) – **complex*16** 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 F07TSF (ZTRTRS).
- 11: LDX – INTEGER *Input*
On entry: the first dimension of the array X as declared in the (sub)program from which F07TVF (ZTRRFS) is called.
Constraint: $\text{LDX} \geq \max(1, N)$.
- 12: FERR(NRHS) – **double precision** array *Output*
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(NRHS) – **double precision** array *Output*
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($2 \times N$) – **complex*16** array *Workspace*
- 15: RWORK(N) – **double precision** array *Workspace*
- 16: 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 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 F07TVF (ZTRRFS), for each right-hand side, involves solving a number of systems of linear equations of the form $Ax = b$ or $A^H x = b$; the number is usually 5 and never more than 11. Each solution involves approximately $4n^2$ real floating-point operations.

The real analogue of this routine is F07THF (DTRRFS).

9 Example

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

$$A = \begin{pmatrix} 4.78 + 4.56i & 0.00 + 0.00i & 0.00 + 0.00i & 0.00 + 0.00i \\ 2.00 - 0.30i & -4.11 + 1.25i & 0.00 + 0.00i & 0.00 + 0.00i \\ 2.89 - 1.34i & 2.36 - 4.25i & 4.15 + 0.80i & 0.00 + 0.00i \\ -1.89 + 1.15i & 0.04 - 3.69i & -0.02 + 0.46i & 0.33 - 0.26i \end{pmatrix}$$

and

$$B = \begin{pmatrix} -14.78 - 32.36i & -18.02 + 28.46i \\ 2.98 - 2.14i & 14.22 + 15.42i \\ -20.96 + 17.06i & 5.62 + 35.89i \\ 9.54 + 9.91i & -16.46 - 1.73i \end{pmatrix}.$$

9.1 Program Text

```
*      F07TVF Example Program Text
*      Mark 16 Release. NAG Copyright 1993.
*      .. 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 ..
      COMPLEX *16      A(LDA,NMAX), B(LDB,NRHMAX), WORK(2*NMAX),
+                     X(LDX,NMAX)
      DOUBLE PRECISION BERR(NRHMAX), FERR(NRHMAX), RWORK(NMAX)
      CHARACTER        CLABS(1), RLABS(1)
*      .. External Subroutines ..
      EXTERNAL         F06TFF, X04DBF, ZTRRFS, ZTRTRS
*      .. Executable Statements ..
      WRITE (NOUT,*) 'F07TVF 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 F06TFF('General',N,NRHS,B,LDB,X,LDX)
*
*          Compute solution in the array X
*
      CALL ZTRTRS(UPLO,TRANS,DIAG,N,NRHS,A,LDA,X,LDX,INFO)
*
*          Compute backward errors and estimated bounds on the
*          forward errors
*
      CALL ZTRRFS(UPLO,TRANS,DIAG,N,NRHS,A,LDA,B,LDB,X,LDX,FERR,BERR,
+               WORK,RWORK,INFO)
*
```

```

*      Print solution
*
      WRITE (NOUT,*)
      IFAIL = 0
      CALL X04DBF('General',' ',N,NRHS,X,LDX,'Bracketed','F7.4',
+               'Solution(s)','Integer',RLABS,'Integer',CLABS,80,0,
+               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 ((5X,1P,4(E11.1,7X)))
      END

```

9.2 Program Data

F07TVF Example Program Data

```

4 2                                     :Values of N and NRHS
'L'                                   :Value of UPLO
( 4.78, 4.56)
( 2.00,-0.30) (-4.11, 1.25)
( 2.89,-1.34) ( 2.36,-4.25) ( 4.15, 0.80)
(-1.89, 1.15) ( 0.04,-3.69) (-0.02, 0.46) ( 0.33,-0.26) :End of matrix A
(-14.78,-32.36) (-18.02, 28.46)
( 2.98, -2.14) ( 14.22, 15.42)
(-20.96, 17.06) ( 5.62, 35.89)
( 9.54, 9.91) (-16.46, -1.73)           :End of matrix B

```

9.3 Program Results

F07TVF Example Program Results

Solution(s)

```

          1          2
1 (-5.0000,-2.0000) ( 1.0000, 5.0000)
2 (-3.0000,-1.0000) (-2.0000,-2.0000)
3 ( 2.0000, 1.0000) ( 3.0000, 4.0000)
4 ( 4.0000, 3.0000) ( 4.0000,-3.0000)

```

Backward errors (machine-dependent)

```

4.2E-17          3.5E-17

```

Estimated forward error bounds (machine-dependent)

```

2.9E-14          3.2E-14

```