

# NAG Library Routine Document

## F07VVF (ZTBFRS)

**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

F07VVF (ZTBFRS) returns error bounds for the solution of a complex triangular band system of linear equations with multiple right-hand sides,  $AX = B$ ,  $A^T X = B$  or  $A^H X = B$ .

### 2 Specification

SUBROUTINE F07VVF (UPLO, TRANS, DIAG, N, KD, NRHS, AB, LDAB, B, LDB, X, &  
LDX, FERR, BERR, WORK, RWORK, INFO)

INTEGER N, KD, NRHS, LDAB, LDB, LDX, INFO  
REAL (KIND=nag\_wp) FERR(NRHS), BERR(NRHS), RWORK(N)  
COMPLEX (KIND=nag\_wp) AB(LDAB,\*), B(LDB,\*), X(LDX,\*), WORK(2\*N)  
CHARACTER(1) UPLO, TRANS, DIAG

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

### 3 Description

F07VVF (ZTBFRS) returns the backward errors and estimated bounds on the forward errors for the solution of a complex triangular band 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 F07VVF (ZTBFRS) 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*  $\beta$ . 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:* 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'  
 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: 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:*  $KD \geq 0$ .
- 6: NRHS – INTEGER *Input*  
*On entry:*  $r$ , the number of right-hand sides.  
*Constraint:* NRHS  $\geq 0$ .
- 7: AB(LDAB,\*) – COMPLEX (KIND=nag\_wp) array *Input*  
**Note:** the second dimension of the array AB must be at least  $\max(1, N)$ .  
*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 F07VVF (ZTBRFS) is called.  
*Constraint:*  $LDAB \geq KD + 1$ .
- 9: B(LDB,\*) – COMPLEX (KIND=nag\_wp) array *Input*  
**Note:** the second dimension of the array B must be at least  $\max(1, NRHS)$ .  
*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 F07VVF (ZTBRFS) is called.  
*Constraint:*  $LDB \geq \max(1, N)$ .
- 11: X(LDX,\*) – COMPLEX (KIND=nag\_wp) array *Input*  
**Note:** the second dimension of the array X must be at least  $\max(1, NRHS)$ .  
*On entry:* the  $n$  by  $r$  solution matrix  $X$ , as returned by F07VSF (ZTBTRS).
- 12: LDX – INTEGER *Input*  
*On entry:* the first dimension of the array X as declared in the (sub)program from which F07VVF (ZTBRFS) is called.  
*Constraint:*  $LDX \geq \max(1, N)$ .
- 13: FERR(NRHS) – REAL (KIND=nag\_wp) 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  $j = 1, 2, \dots, r$ .
- 14: BERR(NRHS) – REAL (KIND=nag\_wp) array *Output*  
*On exit:* BERR( $j$ ) contains the component-wise backward error bound  $\beta$  for the  $j$ th solution vector, that is, the  $j$ th column of  $X$ , for  $j = 1, 2, \dots, r$ .
- 15: WORK( $2 \times N$ ) – COMPLEX (KIND=nag\_wp) array *Workspace*
- 16: RWORK(N) – REAL (KIND=nag\_wp) array *Workspace*
- 17: 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 =  $-i$ , argument  $i$  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 Parallelism and Performance

F07VVF (ZTBRFS) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F07VVF (ZTBRFS) makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

## 9 Further Comments

A call to F07VVF (ZTBRFS), for each right-hand side, involves solving a number of systems of linear equations of the form  $Ax = b$  or  $A^Hx = b$ ; the number is usually 5 and never more than 11. Each solution involves approximately  $8nk$  real floating-point operations (assuming  $n \gg k$ ).

The real analogue of this routine is F07VHF (DTBRFS).

## 10 Example

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

$$A = \begin{pmatrix} -1.94 + 4.43i & 0.00 + 0.00i & 0.00 + 0.00i & 0.00 + 0.00i \\ -3.39 + 3.44i & 4.12 - 4.27i & 0.00 + 0.00i & 0.00 + 0.00i \\ 1.62 + 3.68i & -1.84 + 5.53i & 0.43 - 2.66i & 0.00 + 0.00i \\ 0.00 + 0.00i & -2.77 - 1.93i & 1.74 - 0.04i & 0.44 + 0.10i \end{pmatrix}$$

and

$$B = \begin{pmatrix} -8.86 - 3.88i & -24.09 - 5.27i \\ -15.57 - 23.41i & -57.97 + 8.14i \\ -7.63 + 22.78i & 19.09 - 29.51i \\ -14.74 - 2.40i & 19.17 + 21.33i \end{pmatrix}.$$

### 10.1 Program Text

```

Program f07vvfe

!      F07VVF Example Program Text

!      Mark 25 Release. NAG Copyright 2014.

!      .. Use Statements ..
Use nag_library, Only: nag_wp, x04dbf, ztbrfs, ztbtrs
!      .. 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 ..
Complex (Kind=nag_wp), Allocatable :: ab(:,,:), b(:,,:), work(:), x(:,:)
Real (Kind=nag_wp), Allocatable   :: berr(:), ferr(:), rwork(:)
Character (1)                   :: clabs(1), rlabs(1)
!      .. Intrinsic Procedures ..
Intrinsic                     :: max, min
!      .. Executable Statements ..
Write (nout,*) 'F07VVF 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),work(2*n),x(ldx,n),berr(nrhs), &
         ferr(nrhs),rwork(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 ztbtrs is f07vsf
Call ztbtrs(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 ztbrfs is f07vvf
Call ztbrfs(uplo,trans,diag,n,kd,nrhs,ab,ldab,b,ldb,x,ldx,ferr,berr, &
           work,rwork,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 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(1:nrhs)
Write (nout,*) 'Estimated forward error bounds (machine-dependent)'
Write (nout,99999) ferr(1:nrhs)

99999 Format ((5X,1P,4(E11.1,7X)))
End Program f07vvfe

```

## 10.2 Program Data

F07VVF Example Program Data

```

4 2 2                                     :Values of N, KD and NRHS
'L'                                       :Value of UPLO
(-1.94, 4.43)
(-3.39, 3.44) ( 4.12,-4.27)
( 1.62, 3.68) (-1.84, 5.53) ( 0.43,-2.66)
              (-2.77,-1.93) ( 1.74,-0.04) ( 0.44, 0.10) :End of matrix A
(-8.86, -3.88) (-24.09, -5.27)
(-15.57,-23.41) (-57.97,  8.14)
(-7.63, 22.78) ( 19.09,-29.51)
(-14.74, -2.40) ( 19.17, 21.33)           :End of matrix B

```

### 10.3 Program Results

F07VVF Example Program Results

Solution(s)

	1	2
1	( 0.0000, 2.0000)	( 1.0000, 5.0000)
2	( 1.0000,-3.0000)	(-7.0000,-2.0000)
3	(-4.0000,-5.0000)	( 3.0000, 4.0000)
4	( 2.0000,-1.0000)	(-6.0000,-9.0000)

Backward errors (machine-dependent)

4.1E-17	4.2E-17
---------	---------

Estimated forward error bounds (machine-dependent)

1.8E-14	2.2E-14
---------	---------

---