NAG Toolbox

nag_lapack_ztbrfs (f07vv)

1 Purpose

nag_lapack_ztbrfs (f07vv) 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 Syntax

```
[ferr, berr, info] = nag_lapack_ztbrfs(uplo, trans, diag, kd, ab, b, x, 'n', n,
'nrhs_p', nrhs_p)
```

```
[ferr, berr, info] = f07vv(uplo, trans, diag, kd, ab, b, x, 'n', n, 'nrhs_p',
nrhs_p)
```

3 Description

nag_lapack_ztbrfs (f07vv) 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 function handles each right-hand side vector (stored as a column of the matrix B) independently, so we describe the function of nag_lapack_ztbrfs (f07vv) in terms of a single right-hand side b and solution x.

Given a computed solution x, the function 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

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

Then the function 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

5.1 Compulsory Input Parameters

1: **uplo** – CHARACTER(1)

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)

Indicates the form of the equations.

trans = 'N'
The equations are of the form $AX = B$.

trans = 'T'
The equations are of the form $A^{\mathrm{T}}X = B$.

trans = 'C'
The equations are of the form $A^{\mathrm{H}}X = B$.

Constraint: **trans** = 'N', 'T' or 'C'.

3: **diag** – CHARACTER(1)

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: **kd** – INTEGER

k_d, the number of superdiagonals of the matrix A if **uplo** = 'U', or the number of subdiagonals if **uplo** = 'L'.

Constraint: **kd** ≥ 0.

5: **ab**$(ldab,:)$ – COMPLEX (KIND=nag_wp) array

The first dimension of the array **ab** must be at least **kd** $+ 1$.

The second dimension of the array **ab** must be at least $\max(1, \mathbf{n})$.

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 $\mathbf{ab}(k_d + 1 + i - j, j)$ 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 $\mathbf{ab}(1 + i - j, j)$ 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.

6: **b**$(ldb,:)$ – COMPLEX (KIND=nag_wp) array

The first dimension of the array **b** must be at least $\max(1, \mathbf{n})$.

The second dimension of the array **b** must be at least $\max(1, \mathbf{nrhs\_p})$.

The n by r right-hand side matrix B.

7: **x**$(ldx,:)$ – COMPLEX (KIND=nag_wp) array

The first dimension of the array **x** must be at least $\max(1, \mathbf{n})$.

The second dimension of the array **x** must be at least $\max(1, \textbf{nrhs\_p})$.

The n by r solution matrix X, as returned by nag_lapack_ztbtrs (f07vs).

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the second dimension of the array **ab**.

n, the order of the matrix A.

Constraint: $\mathbf{n} \geq 0$.

2: **nrhs_p** – INTEGER

Default: the second dimension of the arrays **b**, **x**.

r, the number of right-hand sides.

Constraint: $\textbf{nrhs\_p} \geq 0$.

5.3 Output Parameters

1: **ferr**(**nrhs_p**) – REAL (KIND=nag_wp) array

ferr(j) contains an estimated error bound for the jth solution vector, that is, the jth column of X, for $j = 1, 2, \ldots, r$.

2: **berr**(**nrhs_p**) – REAL (KIND=nag_wp) array

berr(j) contains the component-wise backward error bound β for the jth solution vector, that is, the jth column of X, for $j = 1, 2, \ldots, r$.

3: **info** – INTEGER

info $= 0$ unless the function 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 Further Comments

A call to nag_lapack_ztbrfs (f07vv), for each right-hand side, involves solving a number of systems of linear equations of the form $Ax = b$ or $A^{\mathrm{H}}x = b$; 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 function is nag_lapack_dtbrfs (f07vh).

9 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

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

9.1 Program Text

```
     function f07vv_example

fprintf('f07vv example results\n\n');

% Solve AX=B and get error bounds, where A is complex lower triangular banded
% and stored in triangular/symmetric banded format
kd = nag_int(2);
ab = [-1.94 + 4.43i,   4.12 - 4.27i,  0.43 - 2.66i,  0.44 + 0.10i;
      -3.39 + 3.44i,  -1.84 + 5.53i,  1.74 - 0.04i,  0    + 0i;
       1.62 + 3.68i,  -2.77 - 1.93i,  0    + 0i,     0    + 0i];
b = [ -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];

% Solve
uplo  = 'L';
trans = 'N';
diag  = 'N';
[x, info] = f07vs( ...
                   uplo, trans, diag, kd, ab, b);

% Get error bounds
[ferr, berr, info] = f07vv( ...
                            uplo, trans, diag, kd, ab, b, x);

disp('Solution(s)');
disp(x);

fprintf('Backward errors (machine-dependent)\n   ')
fprintf('%11.1e', berr);
fprintf('\nEstimated forward error bounds (machine-dependent)\n   ')
fprintf('%11.1e', ferr);
fprintf('\n');
```

9.2 Program Results

```
     f07vv example results

Solution(s)
   0.0000 + 2.0000i   1.0000 + 5.0000i
   1.0000 - 3.0000i  -7.0000 - 2.0000i
  -4.0000 - 5.0000i   3.0000 + 4.0000i
   2.0000 - 1.0000i  -6.0000 - 9.0000i
```

```
Backward errors (machine-dependent)
       8.3e-18    6.6e-17
Estimated forward error bounds (machine-dependent)
       1.8e-14    2.2e-14
```
