NAG Toolbox

nag_lapack_dtbrfs (f07vh)

1 Purpose

nag_lapack_dtbrfs (f07vh) returns error bounds for the solution of a real triangular band system of linear equations with multiple right-hand sides, $AX = B$ or $A^{T}X = B$.

2 Syntax

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

```
[ferr, berr, info] = f07vh(uplo, trans, diag, kd, ab, b, x, 'n', n, 'nrhs_p', nrhs_p)
```

3 Description

nag_lapack_dtbrfs (f07vh) returns the backward errors and estimated bounds on the forward errors for the solution of a real triangular band system of linear equations with multiple right-hand sides $AX = B$ or $A^{T}X = B$. 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_dtbrfs (f07vh) 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' or 'C'
The equations are of the form $A^{T}X = B$.

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**$(ldab,:)$ – REAL (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**$(ldb,:)$ – REAL (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**$(ldx,:)$ – REAL (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 $\max(1, \mathbf{nrhs\_p})$.

The n by r solution matrix X, as returned by nag_lapack_dtbtrs (f07ve).

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: $\mathbf{nrhs\_p} \geq 0$.

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_dtbrfs (f07vh), 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 $2nk$ floating-point operations (assuming $n \gg k$).

The complex analogue of this function is nag_lapack_ztbrfs (f07vv).

9 Example

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

$$A = \begin{pmatrix} -4.16 & 0.00 & 0.00 & 0.00 \\ -2.25 & 4.78 & 0.00 & 0.00 \\ 0.00 & 5.86 & 6.32 & 0.00 \\ 0.00 & 0.00 & -4.82 & 0.16 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} -16.64 & -4.16 \\ -13.78 & -16.59 \\ 13.10 & -4.94 \\ -14.14 & -9.96 \end{pmatrix}.$$

9.1 Program Text

```
function f07vh_example

fprintf('f07vh example results\n\n');

% Solve AX=B and compute error bounds, where A is lower triangular banded
% and stored in triangular/symmetric banded format
kd = nag_int(1);
ab = [-4.16, 4.78,  6.32, 0.16;
      -2.25, 5.86, -4.82, 0.00];
b = [-16.64,  -4.16;
     -13.78, -16.59;
      13.10,  -4.94;
     -14.14,  -9.96];

% Solve
uplo  = 'L';
trans = 'N';
diag  = 'N';
[x, info] = f07ve( ...
                   uplo, trans, diag, kd, ab, b);

% Compute error bounds
[ferr, berr, info] = f07vh( ...
                            uplo, trans, diag, kd, ab, b, x);

% Display solution
[ifail] = x04ca( ...
                 'Gen', diag, x, 'Solution(s)');

fprintf('\nBackward 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

```
     f07vh example results

 Solution(s)
           1          2
 1      4.0000     1.0000
 2     -1.0000    -3.0000
 3      3.0000     2.0000
 4      2.0000    -2.0000

Backward errors (machine-dependent)
       4.7e-17    2.5e-17
Estimated forward error bounds (machine-dependent)
       5.4e-14    5.8e-14
```