

NAG Toolbox

nag_lapack_ztprfs (f07uv)

1 Purpose

nag_lapack_ztprfs (f07uv) 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 $A^H X = B$, using packed storage.

2 Syntax

```
[ferr, berr, info] = nag_lapack_ztprfs(uplo, trans, diag, ap, b, x, 'n', n,
'nrhs_p', nrhs_p)
[ferr, berr, info] = f07uv(uplo, trans, diag, ap, b, x, 'n', n, 'nrhs_p',
nrhs_p)
```

3 Description

nag_lapack_ztprfs (f07uv) 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 $A^H X = B$, using packed storage. 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_ztprfs (f07uv) 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

$$(A + \delta A)x = b + \delta b$$

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

Then the function 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

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 $A^T X = B$.

trans = 'C'

The equations are of the form $A^H X = B$.

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

3: **diag** – CHARACTER(1)

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: **ap(:)** – COMPLEX (KIND=nag_wp) array

The dimension of the array **ap** must be at least $\max(1, \mathbf{n} \times (\mathbf{n} + 1)/2)$

The n by n triangular matrix A , packed by columns.

More precisely,

if **uplo** = 'U', the upper triangle of A must be stored with element A_{ij} in $\mathbf{ap}(i + j(j - 1)/2)$ for $i \leq j$;

if **uplo** = 'L', the lower triangle of A must be stored with element A_{ij} in $\mathbf{ap}(i + (2n - j)(j - 1)/2)$ for $i \geq j$.

If **diag** = 'U', the diagonal elements of A are assumed to be 1, and are not referenced; the same storage scheme is used whether **diag** = 'N' or 'U'.

5: **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 .

6: **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 $\max(1, \mathbf{nrhs_p})$.

The n by r solution matrix X , as returned by nag_lapack_ztptrs (f07us).

5.2 Optional Input Parameters

1: **n** – INTEGER

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

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**. (An error is raised if these dimensions are not equal.)

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 j th solution vector, that is, the j th column of X , for $j = 1, 2, \dots, r$.

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

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 $j = 1, 2, \dots, r$.

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_ztprfs (f07uv), 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 $4n^2$ real floating-point operations.

The real analogue of this function is nag_lapack_dtprrfs (f07uh).

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

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

using packed storage for A .

9.1 Program Text

```
function f07uv_example

fprintf('f07uv example results\n\n');

% Solve AX=B and obtain error bounds,
% where A is Complex Lower triangular and packed.
ap = [ 4.78 + 4.56i; 2.00 - 0.30i; 2.89 - 1.34i; -1.89 + 1.15i;
       -4.11 + 1.25i; 2.36 - 4.25i; 0.04 - 3.69i;
       4.15 + 0.80i; -0.02 + 0.46i;
       0.33 - 0.26i];

b = [-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];

% Solve
uplo = 'L';
trans = 'N';
diag = 'N';
[x, info] = f07us( ...
    uplo, trans, diag, ap, b);

% Error bounds
[ferr, berr, info] = f07uv( ...
    uplo, trans, diag, ap, 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

```
f07uv example results

Solution(s)
-5.0000 - 2.0000i  1.0000 + 5.0000i
-3.0000 - 1.0000i  -2.0000 - 2.0000i
2.0000 + 1.0000i   3.0000 + 4.0000i
4.0000 + 3.0000i   4.0000 - 3.0000i
```

Backward errors (machine-dependent)
 $6.2\text{e-}17$ $3.5\text{e-}17$
Estimated forward error bounds (machine-dependent)
 $3.0\text{e-}14$ $3.2\text{e-}14$
