NAG Toolbox

nag_lapack_zgbtrs (f07bs)

1 Purpose

nag_lapack_zgbtrs (f07bs) solves a complex band system of linear equations with multiple right-hand sides,

$$AX = B, \quad A^{\mathrm{T}}X = B \quad \text{or} \quad A^{\mathrm{H}}X = B,$$

where A has been factorized by nag_lapack_zgbtrf (f07br).

2 Syntax

```
[b, info] = nag_lapack_zgbtrs(trans, kl, ku, ab, ipiv, b, 'n', n, 'nrhs_p', nrhs_p)
[b, info] = f07bs(trans, kl, ku, ab, ipiv, b, 'n', n, 'nrhs_p', nrhs_p)
```

3 Description

nag_lapack_zgbtrs (f07bs) is used to solve a complex band system of linear equations $AX = B$, $A^{\mathrm{T}}X = B$ or $A^{\mathrm{H}}X = B$, the function must be preceded by a call to nag_lapack_zgbtrf (f07br) which computes the LU factorization of A as $A = PLU$. The solution is computed by forward and backward substitution.

If **trans** = 'N', the solution is computed by solving $PLY = B$ and then $UX = Y$.

If **trans** = 'T', the solution is computed by solving $U^{\mathrm{T}}Y = B$ and then $L^{\mathrm{T}}P^{\mathrm{T}}X = Y$.

If **trans** = 'C', the solution is computed by solving $U^{\mathrm{H}}Y = B$ and then $L^{\mathrm{H}}P^{\mathrm{T}}X = Y$.

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: **trans** – CHARACTER(1)

Indicates the form of the equations.

trans = 'N'
$AX = B$ is solved for X.

trans = 'T'
$A^{\mathrm{T}}X = B$ is solved for X.

trans = 'C'
$A^{\mathrm{H}}X = B$ is solved for X.

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

2: **kl** – INTEGER

k_l, the number of subdiagonals within the band of the matrix A.

Constraint: **kl** ≥ 0.

3: **ku** – INTEGER

k_u, the number of superdiagonals within the band of the matrix A.

Constraint: **ku** ≥ 0.

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

The first dimension of the array **ab** must be at least $2 \times$ **kl** $+$ **ku** $+ 1$.

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

The LU factorization of A, as returned by nag_lapack_zgbtrf (f07br).

5: **ipiv**$(:)$ – INTEGER array

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

The pivot indices, as returned by nag_lapack_zgbtrf (f07br).

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.

5.2 Optional Input Parameters

1: **n** – INTEGER

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

n, the order of the matrix A.

Constraint: **n** ≥ 0.

2: **nrhs_p** – INTEGER

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

r, the number of right-hand sides.

Constraint: **nrhs_p** ≥ 0.

5.3 Output Parameters

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

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

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

The n by r solution matrix X.

2: **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

For each right-hand side vector b, the computed solution x is the exact solution of a perturbed system of equations $(A + E)x = b$, where

$$|E| \leq c(k)\epsilon|L||U|,$$

$c(k)$ is a modest linear function of $k = k_l + k_u + 1$, and ϵ is the ***machine precision***. This assumes $k \ll n$.

If $\hat{x}$ is the true solution, then the computed solution x satisfies a forward error bound of the form

$$\frac{\|x - \hat{x}\|_\infty}{\|x\|_\infty} \leq c(k)\,\mathrm{cond}(A, x)\epsilon$$

where $\mathrm{cond}(A, x) = \left\| |A^{-1}||A||x| \right\|_\infty / \|x\|_\infty \leq \mathrm{cond}(A) = \left\| |A^{-1}||A| \right\|_\infty \leq \kappa_\infty(A)$.

Note that $\mathrm{cond}(A, x)$ can be much smaller than $\mathrm{cond}(A)$, and $\mathrm{cond}(A^{\mathrm{H}})$ (which is the same as $\mathrm{cond}(A^{\mathrm{T}})$) can be much larger (or smaller) than $\mathrm{cond}(A)$.

Forward and backward error bounds can be computed by calling nag_lapack_zgbrfs (f07bv), and an estimate for $\kappa_\infty(A)$ can be obtained by calling nag_lapack_zgbcon (f07bu) with **norm_p** = 'I'.

8 Further Comments

The total number of real floating-point operations is approximately $8n(2k_l + k_u)r$, assuming $n \gg k_l$ and $n \gg k_u$.

This function may be followed by a call to nag_lapack_zgbrfs (f07bv) to refine the solution and return an error estimate.

The real analogue of this function is nag_lapack_dgbtrs (f07be).

9 Example

This example solves the system of equations $AX = B$, where

$$A = \begin{pmatrix} -1.65 + 2.26i & -2.05 - 0.85i & 0.97 - 2.84i & 0.00 + 0.00i \\ 0.00 + 6.30i & -1.48 - 1.75i & -3.99 + 4.01i & 0.59 - 0.48i \\ 0.00 + 0.00i & -0.77 + 2.83i & -1.06 + 1.94i & 3.33 - 1.04i \\ 0.00 + 0.00i & 0.00 + 0.00i & 4.48 - 1.09i & -0.46 - 1.72i \end{pmatrix}$$

and

$$B = \begin{pmatrix} -1.06 + 21.50i & 12.85 + 2.84i \\ -22.72 - 53.90i & -70.22 + 21.57i \\ 28.24 - 38.60i & -20.7 - 31.23i \\ -34.56 + 16.73i & 26.01 + 31.97i \end{pmatrix}.$$

Here A is nonsymmetric and is treated as a band matrix, which must first be factorized by nag_lapack_zgbtrf (f07br).

9.1 Program Text

```
function f07bs_example

fprintf('f07bs example results\n\n');

m = nag_int(4);
kl = nag_int(1);
ku = nag_int(2);
ab = [ 0    + 0i,      0    +  0i,     0    + 0i,     0    + 0i;
       0    + 0i,      0    +  0i,     0.97 - 2.84i,  0.59 - 0.48i;
       0    + 0i,     -2.05 -  0.85i, -3.99 + 4.01i,  3.33 - 1.04i;
      -1.65 + 2.26i,  -1.48 -  1.75i, -1.06 + 1.94i, -0.46 - 1.72i;
       0    + 6.3i,   -0.77 +  2.83i,  4.48 - 1.09i,  0    + 0i];
```

```
b = [ -1.06 + 21.5i,  12.85 +  2.84i;
     -22.72 - 53.9i, -70.22 + 21.57i;
      28.24 - 38.6i, -20.73 -  1.23i;
     -34.56 + 16.73i, 26.01 + 31.97i];

% Factorize
[abf, ipiv, info] = f07br( ...
                           m, kl, ku, ab);

%Solve
trans = 'N';
[x, info] = f07bs( ...
                  trans, kl, ku, abf, ipiv, b);

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

9.2 Program Results

```
     f07bs example results

Solution(s)
  -3.0000 + 2.0000i   1.0000 + 6.0000i
   1.0000 - 7.0000i  -7.0000 - 4.0000i
  -5.0000 + 4.0000i   3.0000 + 5.0000i
   6.0000 - 8.0000i  -8.0000 + 2.0000i
```