NAG Toolbox

nag_linsys_complex_band_solve (f04cb)

1 Purpose

nag_linsys_complex_band_solve (f04cb) computes the solution to a complex system of linear equations $AX = B$, where A is an n by n band matrix, with k_l subdiagonals and k_u superdiagonals, and X and B are n by r matrices. An estimate of the condition number of A and an error bound for the computed solution are also returned.

2 Syntax

```
[ab, ipiv, b, rcond, errbnd, ifail] = nag_linsys_complex_band_solve(kl, ku, ab,
b, 'n', n, 'nrhs_p', nrhs_p)
```

```
[ab, ipiv, b, rcond, errbnd, ifail] = f04cb(kl, ku, ab, b, 'n', n, 'nrhs_p',
nrhs_p)
```

3 Description

The LU decomposition with partial pivoting and row interchanges is used to factor A as $A = PLU$, where P is a permutation matrix, L is the product of permutation matrices and unit lower triangular matrices with k_l subdiagonals, and U is upper triangular with $(k_l + k_u)$ superdiagonals. The factored form of A is then used to solve the system of equations $AX = B$.

4 References

Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J J, Du Croz J J, Greenbaum A, Hammarling S, McKenney A and Sorensen D (1999) *LAPACK Users' Guide* (3rd Edition) SIAM, Philadelphia http://www.netlib.org/lapack/lug

Higham N J (2002) *Accuracy and Stability of Numerical Algorithms* (2nd Edition) SIAM, Philadelphia

5 Parameters

5.1 Compulsory Input Parameters

1: **kl** – INTEGER

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

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

2: **ku** – INTEGER

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

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

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

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

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

The n by n matrix A.

The matrix is stored in rows $k_l + 1$ to $2k_l + k_u + 1$; the first k_l rows need not be set, more precisely, the element A_{ij} must be stored in

$$\mathbf{ab}(k_l + k_u + 1 + i - j, j) = A_{ij} \quad \text{for } \max(1, j - k_u) \leq i \leq \min(n, j + k_l).$$

See Section 9 for further details.

4: $\mathbf{b}(ldb,:)$ – 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 matrix of right-hand sides B.

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the first dimension of the array **b** and the second dimension of the array **ab**.

The number of linear equations n, i.e., the order of the matrix A.

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

2: **nrhs_p** – INTEGER

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

The number of right-hand sides r, i.e., the number of columns of the matrix B.

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

5.3 Output Parameters

1: $\mathbf{ab}(ldab,:)$ – COMPLEX (KIND=nag_wp) array

The first dimension of the array **ab** will be $2 \times \mathbf{kl} + \mathbf{ku} + 1$.

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

If $\mathbf{ifail} \geq 0$, **ab** stores details of the factorization.

The upper triangular band matrix U, with $k_l + k_u$ superdiagonals, is stored in rows 1 to $k_l + k_u + 1$ of the array, and the multipliers used to form the matrix L are stored in rows $k_l + k_u + 2$ to $2k_l + k_u + 1$.

2: $\mathbf{ipiv}(\mathbf{n})$ – INTEGER array

If $\mathbf{ifail} \geq 0$, the pivot indices that define the permutation matrix P; at the ith step row i of the matrix was interchanged with row $\mathbf{ipiv}(i)$. $\mathbf{ipiv}(i) = i$ indicates a row interchange was not required.

3: $\mathbf{b}(ldb,:)$ – 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 $\max(1, \mathbf{nrhs\_p})$.

If $\mathbf{ifail} = 0$ or $\mathbf{n} + 1$, the n by r solution matrix X.

4: **rcond** – REAL (KIND=nag_wp)

If $\mathbf{ifail} \geq 0$, an estimate of the reciprocal of the condition number of the matrix A, computed as $\mathbf{rcond} = \left(\|A\|_1 \|A^{-1}\|_1 \right)$.

5: **errbnd** – REAL (KIND=nag_wp)

If **ifail** $= 0$ or $\mathbf{n} + 1$, an estimate of the forward error bound for a computed solution $\hat{x}$, such that $\|\hat{x} - x\|_1 / \|x\|_1 \leq$ **errbnd**, where $\hat{x}$ is a column of the computed solution returned in the array **b** and x is the corresponding column of the exact solution X. If **rcond** is less than ***machine precision***, then **errbnd** is returned as unity.

6: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail > 0 and **ifail** $\leq$ **n**

Diagonal element $\langle$*value*$\rangle$ of the upper triangular factor is zero. The factorization has been completed, but the solution could not be computed.

ifail $= \mathbf{n} + 1$ (*warning*)

A solution has been computed, but **rcond** is less than ***machine precision*** so that the matrix A is numerically singular.

ifail $= -1$

Constraint: **n** ≥ 0.

ifail $= -2$

Constraint: **kl** ≥ 0.

ifail $= -3$

Constraint: **ku** ≥ 0.

ifail $= -4$

Constraint: **nrhs_p** ≥ 0.

ifail $= -6$

Constraint: $ldab \geq 2 \times$ **kl** $+$ **ku** $+ 1$.

ifail $= -9$

Constraint: $ldb \geq \max(1, \mathbf{n})$.

ifail $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

ifail $= -999$

Dynamic memory allocation failed.

The double allocatable memory required is ***n****, and the complex allocatable memory required is* $2 \times$ **n**. *In this case the factorization and the solution* X *have been computed, but* ***rcond*** *and* ***errbnd*** *have not been computed.*

7 Accuracy

The computed solution for a single right-hand side, $\hat{x}$, satisfies an equation of the form

$$(A+E)\hat{x} = b,$$

where

$$\|E\|_1 = O(\epsilon)\|A\|_1$$

and ϵ is the ***machine precision***. An approximate error bound for the computed solution is given by

$$\frac{\|\hat{x}-x\|_1}{\|x\|_1} \leq \kappa(A)\frac{\|E\|_1}{\|A\|_1},$$

where $\kappa(A) = \left\|A^{-1}\right\|_1 \|A\|_1$, the condition number of A with respect to the solution of the linear equations. nag_linsys_complex_band_solve (f04cb) uses the approximation $\|E\|_1 = \epsilon\|A\|_1$ to estimate **errbnd**. See Section 4.4 of Anderson *et al.* (1999) for further details.

8 Further Comments

The band storage scheme for the array **ab** is illustrated by the following example, when $n = 6$, $k_l = 1$, and $k_u = 2$. Storage of the band matrix A in the array **ab**:

$$\begin{array}{cccccc}
* & * & * & + & + & + \\
* & * & a_{13} & a_{24} & a_{35} & a_{46} \\
* & a_{12} & a_{23} & a_{34} & a_{45} & a_{56} \\
a_{11} & a_{22} & a_{33} & a_{44} & a_{55} & a_{66} \\
a_{21} & a_{32} & a_{43} & a_{54} & a_{65} & *
\end{array}$$

Array elements marked $*$ need not be set and are not referenced by the function. Array elements marked + need not be set, but are defined on exit from the function and contain the elements u_{14}, u_{25} and u_{36}.

The total number of floating-point operations required to solve the equations $AX = B$ depends upon the pivoting required, but if $n \gg k_l + k_u$ then it is approximately bounded by $O(nk_l(k_l + k_u))$ for the factorization and $O(n(2k_l + k_u), r)$ for the solution following the factorization. The condition number estimation typically requires between four and five solves and never more than eleven solves, following the factorization.

In practice the condition number estimator is very reliable, but it can underestimate the true condition number; see Section 15.3 of Higham (2002) for further details.

The real analogue of nag_linsys_complex_band_solve (f04cb) is nag_linsys_real_band_solve (f04bb).

9 Example

This example solves the equations

$$AX = B,$$

where A is the band matrix

$$A = \begin{pmatrix} -1.65+2.26i & -2.05-0.85i & 0.97-2.84i & 0 \\ 0.00+6.30i & -1.48-1.75i & -3.99+4.01i & 0.59-0.48i \\ 0 & -0.77+2.83i & -1.06+1.94i & 3.33-1.04i \\ 0 & 0 & 4.48-1.09i & -0.46-1.72i \end{pmatrix}$$

and

$$B = \begin{pmatrix} -1.06+21.50i & 12.85+2.84i \\ -22.72-53.90i & -70.22+21.57i \\ 28.24-38.60i & -20.73-1.23i \\ -34.56+16.73i & 26.01+31.97i \end{pmatrix}.$$

An estimate of the condition number of A and an approximate error bound for the computed solutions are also printed.

9.1 Program Text

```
function f04cb_example

fprintf('f04cb example results\n\n');

% Solve complex Ax = b for banded A with error bound and condition number
kl = nag_int(1);
ku = nag_int(2);
cz = complex(0);
ab = [ cz,              cz,              cz,              cz;
       cz,              cz,              0.97 - 2.84i,  0.59 - 0.48i;
       cz,             -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    +  6.30i,  -0.77 +  2.83i,  4.48 - 1.09i,  cz         ];
b = [ -1.06 + 21.50i,  12.85 +  2.84i;
     -22.72 - 53.90i, -70.22 + 21.57i;
      28.24 - 38.60i, -20.73 -  1.23i;
     -34.56 + 16.73i,  26.01 + 31.97i];

[ab, ipiv, x, rcond, errbnd, ifail] = ...
  f04cb(kl, ku, ab, b);

disp('Solution');
disp(x);
disp('Estimate of condition number');
fprintf('%10.1f\n\n',1/rcond);
disp('Estimate of error bound for computed solutions');
fprintf('%10.1e\n\n',errbnd);
```

9.2 Program Results

```
f04cb example results

Solution
  -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

Estimate of condition number
     104.2

Estimate of error bound for computed solutions
   1.2e-14
```