NAG Toolbox for MATLAB

f11jb

1 Purpose

f11jb solves a system of linear equations involving the incomplete Cholesky preconditioning matrix generated by f11ja.

2 Syntax

```
[x, ifail] = f11jb(a, irow, icol, ipiv, istr, check, y, 'n', n, 'la', la)
```

3 Description

f11jb solves a system of linear equations

$$Mx = y$$

involving the preconditioning matrix $M = PLDL^{\mathrm{T}}P^{\mathrm{T}}$, corresponding to an incomplete Cholesky decomposition of a sparse symmetric matrix stored in symmetric co-ordinate storage (SCS) format (see Section 2.1.2 in the F11 Chapter Introduction), as generated by f11ja.

In the above decomposition L is a lower triangular sparse matrix with unit diagonal, D is a diagonal matrix and P is a permutation matrix. L and D are supplied to f11jb through the matrix

$$C = L + D^{-1} - I$$

which is a lower triangular **n** by **n** sparse matrix, stored in SCS format, as returned by f11ja. The permutation matrix P is returned from f11ja via the array **ipiv**.

It is envisaged that a common use of f11jb will be to carry out the preconditioning step required in the application of f11ge to sparse symmetric linear systems. f11jb is used for this purpose by the Black Box function f11jc.

f11jb may also be used in combination with f11ja to solve a sparse symmetric positive-definite system of linear equations directly (see Section 8.4 of the document for f11ja). This use of f11jb is demonstrated in Section 9.

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **a**(**la**) – **double array**

The values returned in the array **a** by a previous call to f11da.

2: **irow**(**la**) – **int32 array**

3: **icol**(**la**) – **int32 array**

4: **ipiv**(**n**) – **int32 array**

5: **istr**(**n** + **1**) – **int32 array**

The values returned in arrays **irow**, **icol**, **ipiv** and **istr** by a previous call to f11ja.

6: **check – string**

Specifies whether or not the input data should be checked.

check = 'C'

Checks are carried out on the values of **n**, **irow**, **icol**, **ipiv** and **istr**.

check = 'N'

None of these checks are carried out.

See also Section 8.2.

Constraint: **check** = 'C' or 'N'.

7: **y(n) – double array**

The right-hand side vector *y*.

5.2 Optional Input Parameters

1: **n – int32 scalar**

Default: The dimension of the arrays **ipiv**, **y**, **x**. (An error is raised if these dimensions are not equal.)

n, the order of the matrix *M*. This **must** be the same value as was supplied in the preceding call to f11ja.

Constraint: **n** ≥ 1.

2: **la – int32 scalar**

Default: The dimension of the arrays **a**, **irow**, **icol**. (An error is raised if these dimensions are not equal.)

This **must** be the same value as was supplied in the preceding call to f11ja.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **x(n) – double array**

The solution vector *x*.

2: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail = 1

On entry, **check** $\neq$ 'C' or 'N'.

ifail = 2

On entry, **n** < 1.

ifail $= 3$

On entry, the SCS representation of the preconditioning matrix M is invalid. Further details are given in the error message. Check that the call to f11jb has been preceded by a valid call to f11ja and that the arrays **a**, **irow**, **icol**, **ipiv** and **istr** have not been corrupted between the two calls.

7 Accuracy

The computed solution x is the exact solution of a perturbed system of equations $(M + \delta M)x = y$, where

$$|\delta M| \leq c(n)\epsilon P|L||D|\left|L^{\mathrm{T}}\right|P^{\mathrm{T}},$$

$c(n)$ is a modest linear function of n, and ϵ is the ***machine precision***.

8 Further Comments

8.1 Timing

The time taken for a call to f11jb is proportional to the value of **nnzc** returned from f11ja.

8.2 Use of check

It is expected that a common use of f11jb will be to carry out the preconditioning step required in the application of f11ge to sparse symmetric linear systems. In this situation f11jb is likely to be called many times with the same matrix M. In the interests of both reliability and efficiency, you are recommended to set **check** to 'C' for the first of such calls, and to 'N' for all subsequent calls.

9 Example

```
a = [4;
     -1;
     6;
     1;
     2;
     3;
     2;
     4;
     1;
     2;
     6;
     -4;
     1;
     -1;
     6;
     -1;
     -1;
     3;
     1;
     1;
     -1;
     1;
     4;
     0.3333333333333333;
     -0.3333333333333333;
     0.375;
     0.5;
     0.1666666666666667;
     0.25;
     0.6666666666666666;
     -0.125;
     0.5;
     -0.1666666666666667;
     0.2526315789473684;
```

```
     0.375;
     0.25;
     -0.2210526315789474;
     0.3143093465674111;
     0.1666666666666667;
     0.5;
     0.04210526315789474;
     0.1687344913151365;
     0.3655328798185941;
     0.5;
     -0.6666666666666666;
     -0.25;
     -0.2947368421052631;
     -0.002481389578163769;
     0.4448979591836733;
     0.5889423076923074;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0];
irow = [int32(1);
     int32(2);
     int32(2);
     int32(3);
     int32(3);
     int32(4);
     int32(5);
     int32(5);
     int32(6);
     int32(6);
     int32(6);
     int32(7);
     int32(7);
     int32(7);
     int32(7);
     int32(8);
     int32(8);
     int32(8);
     int32(9);
     int32(9);
     int32(9);
     int32(9);
     int32(9);
     int32(1);
```

```
     int32(2);
     int32(2);
     int32(3);
     int32(4);
     int32(5);
     int32(6);
     int32(6);
     int32(6);
     int32(6);
     int32(6);
     int32(6);
     int32(7);
     int32(7);
     int32(7);
     int32(7);
     int32(8);
     int32(8);
     int32(8);
     int32(8);
     int32(8);
     int32(9);
     int32(9);
     int32(9);
     int32(9);
     int32(9);
     int32(9);
     int32(9);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0)];
icol = [int32(1);
     int32(1);
     int32(2);
     int32(2);
     int32(3);
     int32(4);
     int32(1);
     int32(5);
     int32(3);
     int32(4);
     int32(6);
     int32(2);
     int32(5);
     int32(6);
```

```
     int32(7);
     int32(4);
     int32(6);
     int32(8);
     int32(1);
     int32(5);
     int32(6);
     int32(8);
     int32(9);
     int32(1);
     int32(1);
     int32(2);
     int32(3);
     int32(4);
     int32(5);
     int32(1);
     int32(2);
     int32(3);
     int32(4);
     int32(6);
     int32(2);
     int32(5);
     int32(6);
     int32(7);
     int32(4);
     int32(5);
     int32(6);
     int32(7);
     int32(8);
     int32(3);
     int32(4);
     int32(5);
     int32(6);
     int32(7);
     int32(8);
     int32(9);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0)];
ipiv = [int32(4);
     int32(8);
     int32(3);
     int32(7);
```

```
     int32(1);
     int32(6);
     int32(9);
     int32(5);
     int32(2)];
istr = [int32(24);
     int32(25);
     int32(27);
     int32(28);
     int32(29);
     int32(30);
     int32(35);
     int32(39);
     int32(44);
     int32(51)];
check = 'C';
y = [4.1;
     -2.94;
     1.41;
     2.53;
     4.35;
     1.29;
     5.01;
     0.52;
     4.57];
[x, ifail] = f11jb(a, irow, icol, ipiv, istr, check, y)
```

```
x =
    0.7000
    0.1600
    0.5200
    0.7700
    0.2800
    0.2100
    0.9300
    0.2000
    0.9000
ifail =
           0
```