NAG Toolbox for MATLAB

f11mm

1 Purpose

f11mm computes the reciprocal pivot growth factor of an LU factorization of a real sparse matrix in compressed column (Harwell–Boeing) format.

2 Syntax

```
[rpg, ifail] = f11mm(n, icolzp, a, iprm, il, lval, iu, uval)
```

3 Description

f11mm computes the reciprocal pivot growth factor $\max_j\left(\left\|A_j\right\|_\infty/\left\|U_j\right\|_\infty\right)$ from the columns A_j and U_j of an LU factorization of the matrix A, $P_rAP_c = LU$ where P_r is a row permutation matrix, P_c is a column permutation matrix, L is unit lower triangular and U is upper triangular as computed by f11me.

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **n – int32 scalar**

n, the order of the matrix A.

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

2: **icolzp**$(*)$ **– int32 array**

Note: the dimension of the array **icolzp** must be at least $\mathbf{n}+1$.

icolzp(i) contains the index in A of the start of a new column. See Section 2.1.3 in the F11 Chapter Introduction.

3: **a**$(*)$ **– double array**

Note: the dimension of the array **a** must be at least **icolzp**$(\mathbf{n}+1)-1$, the number of nonzeros of the sparse matrix A.

The array of nonzero values in the sparse matrix A.

4: **iprm**$(7\times\mathbf{n})$ **– int32 array**

The column permutation which defines P_c, the row permutation which defines P_r, plus associated data structures as computed by f11me.

5: **il**$(*)$ **– int32 array**

Note: the dimension of the array **il** must be at least as large as the dimension of the array of the same name in f11me.

Records the sparsity pattern of matrix L as computed by f11me.

6: **lval**$(*)$ **– double array**

Note: the dimension of the array **lval** must be at least as large as the dimension of the array of the same name in f11me.

Records the nonzero values of matrix L and some nonzero values of matrix U as computed by f11me.

7: **iu**$(*)$ **– int32 array**

Note: the dimension of the array **iu** must be at least as large as the dimension of the array of the same name in f11me.

Records the sparsity pattern of matrix U as computed by f11me.

8: **uval**$(*)$ **– double array**

Note: the dimension of the array **uval** must be at least as large as the dimension of the array of the same name in f11me.

Records some nonzero values of matrix U as computed by f11me.

5.2 Optional Input Parameters

None.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **rpg – double scalar**

The reciprocal pivot growth factor $\max_j\left(\|A_j\|_\infty/\|U_j\|_\infty\right)$. If the reciprocal pivot growth factor is much less than 1, the stability of the LU factorization may be poor.

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, $\mathbf{n} < 0$.

ifail $= 2$

Ill-defined column permutations in array **iprm**. Internal checks have revealed that the **iprm** array is corrupted.

ifail $= 301$

Unable to allocate required internal workspace.

7 Accuracy

Not applicable.

8 Further Comments

If the reciprocal pivot growth factor, **rpg**, is much less than 1, then the factorization of the matrix A could be poor. This means that using the factorization to obtain solutions to a linear system, forward error bounds and estimates of the condition number could be unreliable. Consider increasing the **thresh** parameter in the call to f11me.

9 Example

```
n = int32(5);
icolzp = [int32(1);
     int32(3);
     int32(5);
     int32(7);
     int32(9);
     int32(12)];
a = [2;
     4;
     1;
     -2;
     1;
     1;
     -1;
     1;
     1;
     2;
     3];
iprm = [int32(1);
     int32(0);
     int32(4);
     int32(3);
     int32(2);
     int32(4);
     int32(3);
     int32(1);
     int32(2);
     int32(0);
     int32(2);
     int32(0);
     int32(8);
     int32(6);
     int32(4);
     int32(4);
     int32(2);
     int32(11);
     int32(8);
     int32(6);
     int32(1);
     int32(2);
     int32(3);
     int32(4);
     int32(5);
     int32(2);
     int32(2);
     int32(2);
     int32(2);
     int32(1);
     int32(1);
     int32(1);
     int32(1);
     int32(2);
     int32(0)];
il = [int32(0);
     int32(1);
     int32(2);
     int32(3);
```

```
      int32(-1);
      int32(-1);
      int32(1);
      int32(2);
      int32(3);
      int32(5);
      int32(-1);
      int32(0);
      int32(1);
      int32(2);
      int32(3);
      int32(3);
      int32(3);
      int32(0);
      int32(2);
      int32(4);
      int32(6);
      int32(14);
      int32(8);
      int32(2);
      int32(4);
      int32(6);
      int32(8);
      int32(15);
      int32(0);
      int32(2);
      int32(4);
      int32(6);
      int32(8);
      int32(-1);
      int32(2);
      int32(4);
      int32(6);
      int32(8);
      int32(10);
      int32(0);
      int32(4);
      int32(1);
      int32(4);
      int32(2);
      int32(4);
      int32(3);
      int32(4);
      int32(3);
      int32(0);
      int32(3);
      int32(0);
      int32(1);
      int32(0);
      int32(0);
      int32(0);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
```

```
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1)];
lval = [-2;
      -0.5;
      4;
      0.5;
      2;
      0.5;
      -1;
      0.5;
      1;
      -1;
      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;
        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;
        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];
iu = [int32(0);
      int32(0);
      int32(0);
      int32(2);
      int32(3);
      int32(-1);
      int32(0);
      int32(0);
      int32(2);
      int32(3);
      int32(4);
      int32(1);
      int32(0);
      int32(2);
      int32(1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
```

```
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1);
      int32(-1)];
uval = [1;
      3;
      1;
      1;
      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;
      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;
      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;
      0;
      0;
      0;
      0];
[rpg, ifail] = f11mm(n, icolzp, a, iprm, il, lval, iu, uval)
```

```
rpg =
      1
ifail =
           0
```