NAG Library Routine Document

F11MMF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

F11MMF computes the reciprocal pivot growth factor of an LU factorization of a real sparse matrix in compressed column (Harwell–Boeing) format.

2 Specification

```
SUBROUTINE F11MMF (N, ICOLZP, A, IPRM, IL, LVAL, IU, UVAL, RPG, IFAIL)
INTEGER            N, ICOLZP(*), IPRM(7*N), IL(*), IU(*), IFAIL
REAL (KIND=nag_wp) A(*), LVAL(*), UVAL(*), RPG
```

3 Description

F11MMF 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 F11MEF.

4 References

None.

5 Arguments

1: N – INTEGER *Input*

On entry: n, the order of the matrix A.

Constraint: $\mathrm{N} \geq 0$.

2: ICOLZP($*$) – INTEGER array *Input*

Note: the dimension of the array ICOLZP must be at least $\mathrm{N}+1$.

On entry: 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($*$) – REAL (KIND=nag_wp) array *Input*

Note: the dimension of the array A must be at least $\mathrm{ICOLZP}(\mathrm{N}+1)-1$, the number of nonzeros of the sparse matrix A.

On entry: the array of nonzero values in the sparse matrix A.

4: IPRM($7 \times \mathrm{N}$) – INTEGER array *Input*

On entry: the column permutation which defines P_c, the row permutation which defines P_r, plus associated data structures as computed by F11MEF.

5: IL(∗) – INTEGER array *Input*

Note: the dimension of the array IL must be at least as large as the dimension of the array of the same name in F11MEF.

On entry: records the sparsity pattern of matrix L as computed by F11MEF.

6: LVAL(∗) – REAL (KIND=nag_wp) array *Input*

Note: the dimension of the array LVAL must be at least as large as the dimension of the array of the same name in F11MEF.

On entry: records the nonzero values of matrix L and some nonzero values of matrix U as computed by F11MEF.

7: IU(∗) – INTEGER array *Input*

Note: the dimension of the array IU must be at least as large as the dimension of the array of the same name in F11MEF.

On entry: records the sparsity pattern of matrix U as computed by F11MEF.

8: UVAL(∗) – REAL (KIND=nag_wp) array *Input*

Note: the dimension of the array UVAL must be at least as large as the dimension of the array of the same name in F11MEF.

On entry: records some nonzero values of matrix U as computed by F11MEF.

9: RPG – REAL (KIND=nag_wp) *Output*

On exit: 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.

10: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this argument, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL $= 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL $= 1$

On entry, N $= \langle value \rangle$.
Constraint: N ≥ 0.

IFAIL $= 2$

Incorrect column permutations in array IPRM.

IFAIL $= -99$

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

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -399$

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

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

Not applicable.

8 Parallelism and Performance

F11MMF is not threaded in any implementation.

9 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 argument in the call to F11MEF.

10 Example

To compute the reciprocal pivot growth for the factorization of the matrix A, where

$$A = \begin{pmatrix} 2.00 & 1.00 & 0 & 0 & 0 \\ 0 & 0 & 1.00 & -1.00 & 0 \\ 4.00 & 0 & 1.00 & 0 & 1.00 \\ 0 & 0 & 0 & 1.00 & 2.00 \\ 0 & -2.00 & 0 & 0 & 3.00 \end{pmatrix}.$$

In this case, it should be equal to 1.0.

10.1 Program Text

```
    Program f11mmfe

!     F11MMF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: f11mdf, f11mef, f11mmf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Real (Kind=nag_wp), Parameter    :: one = 1.E0_nag_wp
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: flop, rpg, thresh
      Integer                          :: i, ifail, n, nnz, nnzl, nnzu, nzlmx, &
                                          nzlumx, nzumx
      Character (1)                    :: spec
```

```
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:), lval(:), uval(:)
      Integer, Allocatable             :: icolzp(:), il(:), iprm(:),           &
                                          irowix(:), iu(:)
!     .. Executable Statements ..
      Write (nout,*) 'F11MMF Example Program Results'
!     Skip heading in data file
      Read (nin,*)

!     Read order of matrix

      Read (nin,*) n

      Allocate (icolzp(n+1),iprm(7*n))

!     Read the matrix A

      Read (nin,*) icolzp(1:n+1)
      nnz = icolzp(n+1) - 1

      Allocate (a(nnz),lval(8*nnz),uval(8*nnz),il(7*n+8*nnz+4),irowix(nnz),    &
        iu(2*n+8*nnz+1))

      Do i = 1, nnz
        Read (nin,*) a(i), irowix(i)
      End Do

!     Calculate COLAMD permutation

      spec = 'M'

!     ifail: behaviour on error exit
!             =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 0
      Call f11mdf(spec,n,icolzp,irowix,iprm,ifail)

!     Factorise

      thresh = one
      ifail = 0
      nzlmx = 8*nnz
      nzlumx = 8*nnz
      nzumx = 8*nnz

      Call f11mef(n,irowix,a,iprm,thresh,nzlmx,nzlumx,nzumx,il,lval,iu,uval,   &
        nnzl,nnzu,flop,ifail)

!     Calculate reciprocal pivot growth

      ifail = 0
      Call f11mmf(n,icolzp,a,iprm,il,lval,iu,uval,rpg,ifail)

!     Output result

      Write (nout,*)
      Write (nout,*) 'Reciprocal pivot growth'
      Write (nout,'(F7.3)') rpg

    End Program f11mmfe
```

10.2 Program Data

```
F11MMF Example Program Data
  5  N
 1
 3
 5
 7
 9
 12   ICOLZP(I) I=1,..,N+1
```

```
  2.   1
  4.   3
  1.   1
 -2.   5
  1.   2
  1.   3
 -1.   2
  1.   4
  1.   3
  2.   4
  3.   5    A(I), IROWIX(I) I=1,NNZ
```

10.3 Program Results

```
 F11MMF Example Program Results

 Reciprocal pivot growth
  1.000
```