NAG Library Routine Document

F11MFF

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

F11MFF solves a real sparse system of linear equations with multiple right-hand sides given an LU factorization of the sparse matrix computed by F11MEF.

2 Specification

```
SUBROUTINE F11MFF (TRANS, N, IPRM, IL, LVAL, IU, UVAL, NRHS, B, LDB,       &
                   IFAIL)

INTEGER            N, IPRM(7*N), IL(*), IU(*), NRHS, LDB, IFAIL
REAL (KIND=nag_wp) LVAL(*), UVAL(*), B(LDB,*)
CHARACTER(1)       TRANS
```

3 Description

F11MFF solves a real system of linear equations with multiple right-hand sides $AX = B$ or $A^{\mathrm{T}}X = B$, according to the value of the parameter TRANS, where the matrix factorization $P_rAP_c = LU$ corresponds to an LU decomposition of a sparse matrix stored in compressed column (Harwell–Boeing) format, as computed by F11MEF.

In the above decomposition L is a lower triangular sparse matrix with unit diagonal elements and U is an upper triangular sparse matrix; P_r and P_c are permutation matrices.

4 References

None.

5 Parameters

1: TRANS – CHARACTER(1) *Input*

On entry: specifies whether $AX = B$ or $A^{\mathrm{T}}X = B$ is solved.

TRANS = 'N'
    $AX = B$ is solved.

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

Constraint: TRANS = 'N' or 'T'.

2: N – INTEGER *Input*

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

Constraint: N ≥ 0.

3: 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.

4: 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.

5: 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.

6: 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.

7: 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.

8: NRHS – INTEGER *Input*

On entry: $nrhs$, the number of right-hand sides in B.

Constraint: NRHS ≥ 0.

9: B(LDB, $*$) – REAL (KIND=nag_wp) array *Input/Output*

Note: the second dimension of the array B must be at least $\max(1, \mathrm{NRHS})$.

On entry: the N by NRHS right-hand side matrix B.

On exit: the N by NRHS solution matrix X.

10: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F11MFF is called.

Constraint: LDB $\geq \max(1, \mathrm{N})$.

11: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction 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 parameter, 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, TRANS $\neq$ 'N' or 'T',
or N < 0,
or NRHS < 0,
or LDB $< \max(1, \mathrm{N})$.

IFAIL $= 2$

Ill-defined row permutation in array IPRM. Internal checks have revealed that the IPRM array is corrupted.

IFAIL $= 3$

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.

IFAIL $= -99$

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

See Section 3.8 in the Essential Introduction for further information.

IFAIL $= -399$

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

See Section 3.7 in the Essential Introduction for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

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

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

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

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

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

where $\operatorname{cond}(A, x) = \left\||A^{-1}||A||x|\right\|_\infty / \|x\|_\infty \leq \operatorname{cond}(A) = \left\||A^{-1}||A|\right\|_\infty \leq \kappa_\infty(A)$. Note that $\operatorname{cond}(A, x)$ can be much smaller than $\operatorname{cond}(A)$, and $\operatorname{cond}(A^{\mathrm{T}})$ can be much larger (or smaller) than $\operatorname{cond}(A)$.

Forward and backward error bounds can be computed by calling F11MHF, and an estimate for $\kappa_\infty(A)$ can be obtained by calling F11MGF.

8 Parallelism and Performance

F11MFF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F11MFF makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

F11MFF may be followed by a call to F11MHF to refine the solution and return an error estimate.

10 Example

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

$$A = \begin{pmatrix} 2.00 & 1.00 & 0 & 0 & 0 \\ 0 & 0 & 1.00 & -1.00 & 0 \\ 4.00 & 0 & 1.00 & 0 & 1.00 \\ 0 & 0 & 0 & 1.00 & 2.00 \\ 0 & -2.00 & 0 & 0 & 3.00 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 1.56 & 3.12 \\ -0.25 & -0.50 \\ 3.60 & 7.20 \\ 1.33 & 2.66 \\ 0.52 & 1.04 \end{pmatrix}.$$

Here A is nonsymmetric and must first be factorized by F11MEF.

10.1 Program Text

```
    Program f11mffe

!     F11MFF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: f11mdf, f11mef, f11mff, nag_wp, x04caf
!     .. 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, thresh
      Integer                          :: i, ifail, j, ldb, n, nnz, nnzl,      &
                                          nnzu, nrhs, nzlmx, nzlumx, nzumx
      Character (1)                    :: spec, trans
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:), b(:,:), lval(:), uval(:)
      Integer, Allocatable             :: icolzp(:), il(:), iprm(:),           &
                                          irowix(:), iu(:)
!     .. Executable Statements ..
      Write (nout,*) 'F11MFF Example Program Results'
      Flush (nout)
!     Skip heading in data file
      Read (nin,*)

!     Read order of matrix and number of right hand sides

      Read (nin,*) n, nrhs
      ldb = n

      Allocate (b(ldb,nrhs),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

!     Read the right hand sides

      Do j = 1, nrhs
        Read (nin,*) b(1:n,j)
      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)

!     Solve

      trans = 'N'

      ifail = 0
      Call f11mff(trans,n,iprm,il,lval,iu,uval,nrhs,b,ldb,ifail)

!     Output results
      Write (nout,*)
      Flush (nout)

      Call x04caf('G',' ',n,nrhs,b,ldb,'Solutions',ifail)

    End Program f11mffe
```

10.2 Program Data

```
F11MFF Example Program Data
  5 2  N, NRHS
 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
1.56 -.25 3.6 1.33 .52
3.12 -.50 7.2 2.66 1.04 B(I,J) J=1,NRHS I=1,N
```

10.3 Program Results

```
F11MFF Example Program Results

Solutions
            1          2
1      0.7000     1.4000
2      0.1600     0.3200
3      0.5200     1.0400
4      0.7700     1.5400
5      0.2800     0.5600
```