NAG Fortran Library Routine Document

F11MKF

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

F11MKF computes a matrix-matrix or transposed matrix-matrix product involving a real, square, sparse nonsymmetric matrix stored in compressed column (Harwell–Boeing) format.

2 Specification

```
 SUBROUTINE F11MKF (TRANS, N, M, ALPHA, ICOLZP, IROWIX, A, B, LDB, BETA,
1                   C, LDC, IFAIL)
 INTEGER            N, M, ICOLZP(*), IROWIX(*), LDB, LDC, IFAIL
 double precision   ALPHA, A(*), B(LDB,*), BETA, C(LDC,*)
 CHARACTER*1        TRANS
```

3 Description

F11MKF computes either the matrix-matrix product $C \leftarrow \alpha AB + \beta C$, or the transposed matrix-matrix product $C \leftarrow \alpha A^T B + \beta C$, according to the value of the argument TRANS, where A is a real n by n sparse nonsymmetric matrix, of arbitrary sparsity pattern with nnz non-zero elements, B and C are n by m real dense matrices. The matrix A is stored in compressed column (Harwell–Boeing) storage format. The array A stores all non-zero elements of A, while arrays ICOLZP and IROWIX store the compressed column indices and row indices of A respectively.

4 References

None.

5 Parameters

1: TRANS – CHARACTER*1 *Input*

On entry: specifies whether or not the matrix A is transposed:

if TRANS = 'N', then $\alpha AB + \beta C$ is computed;
if TRANS = 'T', then $\alpha A^T B + \beta C$ is computed.

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

2: N – INTEGER *Input*

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

Constraint: N $\geq$ 0.

3: M – INTEGER *Input*

On entry: m, the number of columns of matrices B and C.

Constraint: M $\geq$ 0.

4: ALPHA – ***double precision*** *Input*

On entry: α, the scalar factor in the matrix multiplication.

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

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.

6: IROWIX($*$) – INTEGER array *Input*

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

On entry: the row index array of the sparse matrix A.

7: A($*$) – ***double precision*** array *Input*

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

On entry: the array of non-zero values in the sparse matrix A.

8: B(LDB,$*$) – ***double precision*** array *Input*

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

On entry: the n by m matrix B.

9: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F11MKF is called.

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

10: BETA – ***double precision*** *Input*

On entry: the scalar factor β.

11: C(LDC,$*$) – ***double precision*** array *Input/Output*

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

On entry: the n by m matrix C.

On exit: C is overwritten by $\alpha AB+\beta C$ or $\alpha A^TB+\beta C$ depending on the value of TRANS.

12: LDC – INTEGER *Input*

On entry: the first dimension of the array C as declared in the (sub)program from which F11MKF is called.

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

13: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

On exit: $\mathrm{IFAIL}=0$ unless the routine detects an error (see Section 6).

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, for users 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.**

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 M < 0,
or LDB $< \max(1, \mathrm{N})$,
or LDC $< \max(1, \mathrm{N})$.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

This example program reads in a sparse matrix A and a dense matrix B. It then calls F11MKF to compute the matrix-matrix product $C = AB$ and the transposed matrix-matrix product $C = A^T B$, where

$$A = \begin{pmatrix} 2.00 & 1.00 & 0 & 0 & 0 \\ 0 & 0 & 1.00 & -1.00 & 0 \\ 4.00 & 0 & 1.00 & 0 & 1.00 \\ 0 & 0 & 0 & 1.00 & 2.00 \\ 0 & -2.00 & 0 & 0 & 3.00 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 0.70 & 1.40 \\ 0.16 & 0.32 \\ 0.52 & 1.04 \\ 0.77 & 1.54 \\ 0.28 & 0.56 \end{pmatrix}.$$

9.1 Program Text

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     F11MKF Example Program Text
*     Mark 21 Release. NAG Copyright 2004.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          LA, NMAX
      PARAMETER        (LA=10000,NMAX=1000)
      DOUBLE PRECISION ONE, ZERO
      PARAMETER        (ONE=1.D0,ZERO=0.D0)
      INTEGER          MMAX
      PARAMETER        (MMAX=5)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, J, M, N, NNZ
      CHARACTER        TRANS
*     .. Local Arrays ..
      DOUBLE PRECISION A(LA), B(NMAX,MMAX), C(NMAX,MMAX)
      INTEGER          ICOLZP(NMAX+1), IROWIX(LA)
*     .. External Subroutines ..
      EXTERNAL         F11MKF, X04CAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F11MKF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
*
*     Read order of matrix
```

```
*
      READ (NIN,*) N, M
      IF (N.LE.NMAX .AND. M.LE.MMAX) THEN
*
*        Read the matrix A
*
         DO 20 I = 1, N + 1
            READ (NIN,*) ICOLZP(I)
   20    CONTINUE
         NNZ = ICOLZP(N+1) - 1
         DO 40 I = 1, NNZ
            READ (NIN,*) A(I), IROWIX(I)
   40    CONTINUE
*
*        Read the matrix B
*
         DO 60 J = 1, M
            READ (NIN,*) (B(I,J),I=1,N)
   60    CONTINUE
*
*        Calculate matrix-matrix product
*
         TRANS = 'N'
         IFAIL = 0
         CALL F11MKF(TRANS,N,M,ONE,ICOLZP,IROWIX,A,B,NMAX,ZERO,C,NMAX,
     +               IFAIL)
*
*        Output results
*
         WRITE (NOUT,*)
         CALL X04CAF('G',' ',N,M,C,NMAX,'Matrix-vector product',IFAIL)
*
*        Calculate transposed matrix-matrix product
*
         TRANS = 'T'
         IFAIL = 0
         CALL F11MKF(TRANS,N,M,ONE,ICOLZP,IROWIX,A,B,NMAX,ZERO,C,NMAX,
     +               IFAIL)
*
*        Output results
*
         WRITE (NOUT,*)
         CALL X04CAF('G',' ',N,M,C,NMAX,
     +               'Transposed matrix-vector product',IFAIL)
*
      END IF
      END
```

9.2 Program Data

```
F11MKF Example Program Data
  5 2                 N, M
 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
 0.70 0.16 0.52  0.77 0.28
```

```
1.40 0.32 1.04  1.54 0.56     matrix B
```

9.3 Program Results

```
F11MKF Example Program Results

Matrix-vector product
            1          2
1      1.5600     3.1200
2     -0.2500    -0.5000
3      3.6000     7.2000
4      1.3300     2.6600
5      0.5200     1.0400

Transposed matrix-vector product
            1          2
1      3.4800     6.9600
2      0.1400     0.2800
3      0.6800     1.3600
4      0.6100     1.2200
5      2.9000     5.8000
```