F11XEF – NAG Fortran Library Routine Document

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

Computes a matrix-vector product involving a real sparse symmetric matrix stored in symmetric coordinate storage format.

2 Specification

```
SUBROUTINE F11XEF(N, NNZ, A, IROW, ICOL, CHECK, X, Y, IFAIL)
INTEGER           N, NNZ, IROW(NNZ), ICOL(NNZ), IFAIL
real              A(NNZ), X(N), Y(N)
CHARACTER*1       CHECK
```

3 Description

F11XEF computes the matrix-vector product

$$y = Ax$$

where A is an N by N symmetric sparse matrix, of arbitrary sparsity pattern, stored in symmetric coordinate storage (SCS) format (see Section 2.1.2 of the Chapter Introduction). The array A stores all non-zero elements in the lower triangular part of A, while arrays IROW and ICOL store the corresponding row and column indices respectively.

It is envisaged that a common use of F11XEF will be to compute the matrix-vector product required in the application of F11GBF to sparse symmetric linear systems. An illustration of this usage appears in Section 9 of the document for F11JDF.

4 References

None.

5 Parameters

1: N — INTEGER *Input*

On entry: the order of the matrix A.

Constraint: N ≥ 1.

2: NNZ — INTEGER *Input*

On entry: the number of non-zero elements in the lower triangular part of A.

Constraint: $1 \leq$ NNZ $\leq$ N $\times$ (N+1)/2.

3: A(NNZ) — ***real*** array *Input*

On entry: the non-zero elements in the lower triangular part of the matrix A, ordered by increasing row index, and by increasing column index within each row. Multiple entries for the same row and column indices are not permitted. The routine F11ZBF may be used to order the elements in this way.

4: IROW(NNZ) — INTEGER array *Input*

5: ICOL(NNZ) — INTEGER array *Input*

On entry: the row and column indices of the non-zero elements supplied in A.

Constraints: IROW and ICOL must satisfy the following constraints (which may be imposed by a call to F11ZBF):

$1 \le \text{IROW}(i) \le \text{N}$, $1 \le \text{ICOL}(i) \le \text{IROW}(i)$, for $i = 1, 2, \ldots$,NNZ.
IROW$(i-1) <$ IROW(i), or
IROW$(i-1) =$ IROW(i) and ICOL$(i-1) <$ ICOL(i), for $i = 2, 3, \ldots$,NNZ.

6: CHECK — CHARACTER*1 *Input*

On entry: specifies whether or not the input data should be checked:

if CHECK = 'C', checks are carried out on the values of N, NNZ, IROW and ICOL;
if CHECK = 'N', none of these checks are carried out.

See also Section 8.2.

Constraint: CHECK = 'C' or 'N'.

7: X(N) — ***real*** array *Input*

On entry: the vector x.

8: Y(N) — ***real*** array *Output*

On exit: the vector y.

9: IFAIL — INTEGER *Input/Output*

On entry: IFAIL must be set to 0, −1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

On exit: IFAIL = 0 unless the routine detects an error (see Section 6).

6 Errors 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 detected by the routine:

IFAIL = 1

On entry, CHECK ≠ 'C' or 'N'.

IFAIL = 2

On entry, N < 1,
or NNZ < 1,
or NNZ > N × (N+1)/2.

IFAIL = 3

On entry, the arrays IROW and ICOL fail to satisfy the following constraints:

$1 \le \text{IROW}(i) \le \text{N}$ and $1 \le \text{ICOL}(i) \le \text{IROW}(i)$, for $i = 1, 2, \ldots$,NNZ.
IROW$(i-1) <$ IROW(i), or
IROW$(i-1) =$ IROW(i) and ICOL$(i-1) <$ ICOL(i), for $i = 2, 3, \ldots$,NNZ.

Therefore a non-zero element has been supplied which does not lie in the lower triangular part of A, is out of order, or has duplicate row and column indices. Call F11ZBF to reorder and sum or remove duplicates.

7 Accuracy

The computed vector y satisfies the error bound:

$$\|y - Ax\|_\infty \leq c(n)\epsilon\|A\|_\infty\|x\|_\infty,$$

where $c(n)$ 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 F11XEF is proportional to NNZ.

8.2 Use of CHECK

It is expected that a common use of F11XEF will be to compute the matrix-vector product required in the application of F11GBF to sparse symmetric linear systems. In this situation F11XEF is likely to be called many times with the same matrix A. 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

This example program reads in a symmetric positive-definite sparse matrix A and a vector x. It then calls F11XEF to compute the matrix-vector product $y = Ax$.

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.

```
*     F11XEF Example Program Text
*     Mark 17 Release. NAG Copyright 1995.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          LA, NMAX
      PARAMETER        (LA=10000,NMAX=1000)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, N, NNZ
      CHARACTER        CHECK
*     .. Local Arrays ..
      real             A(LA), X(NMAX), Y(NMAX)
      INTEGER          ICOL(LA), IROW(LA)
*     .. External Subroutines ..
      EXTERNAL         F11XEF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F11XEF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
*
*     Read order of matrix and number of non-zero entries
*
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
         READ (NIN,*) NNZ
*
*     Read the matrix A
*
```

```
         DO 20 I = 1, NNZ
            READ (NIN,*) A(I), IROW(I), ICOL(I)
   20    CONTINUE
*
*     Read the vector x
*
         READ (NIN,*) (X(I),I=1,N)
*
*     Calculate matrix-vector product
*
         CHECK = 'C'
         IFAIL = 0
         CALL F11XEF(N,NNZ,A,IROW,ICOL,CHECK,X,Y,IFAIL)
*
*     Output results
*
         WRITE (NOUT,*) ' Matrix-vector product'
         DO 40 I = 1, N
            WRITE (NOUT,99999) Y(I)
   40    CONTINUE
      END IF
      STOP
*
99999 FORMAT (1X,e16.4)
      END
```

9.2 Program Data

```
F11XEF Example Program Data
  9                   N
 23                   NNZ
  4.   1    1
 -1.   2    1
  6.   2    2
  1.   3    2
  2.   3    3
  3.   4    4
  2.   5    1
  4.   5    5
  1.   6    3
  2.   6    4
  6.   6    6
 -4.   7    2
  1.   7    5
 -1.   7    6
  6.   7    7
 -1.   8    4
 -1.   8    6
  3.   8    8
  1.   9    1
  1.   9    5
 -1.   9    6
  1.   9    8
  4.   9    9          A(I), IROW(I), ICOL(I), I=1,...,NNZ
 0.70 0.16 0.52
 0.77 0.28 0.21
 0.93 0.20 0.90       X(I), I=1,...,N
```

9.3 Program Results

```
F11XEF Example Program Results
 Matrix-vector product
      0.4100E+01
     -0.2940E+01
      0.1410E+01
      0.2530E+01
      0.4350E+01
      0.1290E+01
      0.5010E+01
      0.5200E+00
      0.4570E+01
```
