NAG Library Routine Document

X04CCF

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

X04CCF is an easy-to-use routine to print a real triangular matrix stored in a packed one-dimensional array.

2 Specification

```
SUBROUTINE X04CCF (UPLO, DIAG, N, A, TITLE, IFAIL)

INTEGER            N, IFAIL
REAL (KIND=nag_wp) A(*)
CHARACTER(*)       TITLE
CHARACTER(1)       UPLO, DIAG
```

3 Description

X04CCF prints a real triangular matrix stored in packed form. It is an easy-to-use driver for X04CDF. The routine uses default values for the format in which numbers are printed, for labelling the rows and columns, and for output record length. The matrix must be packed by column.

X04CCF will choose a format code such that numbers will be printed with an F8.4, an F11.4 or a 1PE13.4 format . The F8.4 code is chosen if the sizes of all the matrix elements to be printed lie between 0.001 and 1.0. The F11.4 code is chosen if the sizes of all the matrix elements to be printed lie between 0.001 and 9999.9999. Otherwise the 1PE13.4 code is chosen.

The matrix is printed with integer row and column labels, and with a maximum record length of 80.

The matrix is output to the unit defined by X04ABF.

4 References

None.

5 Parameters

1: UPLO – CHARACTER(1) *Input*

On entry: indicates the type of the matrix to be printed

UPLO = 'L'
The matrix is lower triangular. In this case, the packed array A holds the matrix elements in the following order: $(1,1),(2,1),\ldots,(\mathrm{N},1),(2,2),(3,2),\ldots,(\mathrm{N},2)$, etc.

UPLO = 'U'
The matrix is upper triangular. In this case, the packed array A holds the matrix elements in the following order: $(1,1),(1,2),(2,2),(1,3),(2,3),(3,3),(1,4)$, etc.

Constraint: UPLO = 'L' or 'U'.

2: DIAG – CHARACTER(1) *Input*

On entry: indicates whether the diagonal elements of the matrix are to be printed.

DIAG = 'B'
The diagonal elements of the matrix are not referenced and not printed.

DIAG = 'U'
The diagonal elements of the matrix are not referenced, but are assumed all to be unity, and are printed as such.

DIAG = 'N'
The diagonal elements of the matrix are referenced and printed.

Constraint: DIAG = 'B', 'U' or 'N'.

3: N – INTEGER *Input*

On entry: the order of the matrix to be printed.

If N is less than 1, X04CCF will exit immediately after printing TITLE; no row or column labels are printed.

4: A(∗) – REAL (KIND=nag_wp) array *Input*

Note: the dimension of the array A must be at least $\max(1, \mathrm{N} \times (\mathrm{N}+1)/2)$.

On entry: the matrix to be printed. Note that A must have space for the diagonal elements of the matrix, even if these are not stored.

More precisely,

if UPLO = 'U', the upper triangle of A must be stored with element A_{ij} in $\mathrm{A}(i + j(j-1)/2)$ for $i \leq j$;

if UPLO = 'L', the lower triangle of A must be stored with element A_{ij} in $\mathrm{A}(i + (2n - j)(j-1)/2)$ for $i \geq j$.

If DIAG = 'U', the diagonal elements of A are assumed to be 1, and are not referenced; the same storage scheme is used whether DIAG = 'N' or 'U'.

5: TITLE – CHARACTER(*) *Input*

On entry: a title to be printed above the matrix.

If TITLE = ' ', no title (and no blank line) will be printed.

If TITLE contains more than 80 characters, the contents of TITLE will be wrapped onto more than one line, with the break after 80 characters.

Any trailing blank characters in TITLE are ignored.

6: 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, UPLO $\neq$ 'L' or 'U'.

IFAIL $= 2$

On entry, DIAG $\neq$ 'N', 'U' or 'B'.

7 Accuracy

Not applicable.

8 Further Comments

A call to X04CCF is equivalent to a call to X04CDF with the following argument values:

```
NCOLS = 80
INDENT = 0
LABROW = 'I'
LABCOL = 'I'
FORM = ' '
```

9 Example

The example program calls X04CCF twice, first to print a 4 by 4 lower triangular matrix, and then to print a 5 by 5 upper triangular matrix.

9.1 Program Text

```
    Program x04ccfe

!     X04CCF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, x04ccf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nmax = 5, nout = 6
      Integer, Parameter               :: la = (nmax*(nmax+1))/2
!     .. Local Scalars ..
      Integer                          :: i, ifail
!     .. Local Arrays ..
      Real (Kind=nag_wp)               :: a(la)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: real
!     .. Executable Statements ..
      Write (nout,*) 'X04CCF Example Program Results'

      Write (nout,*)
      Flush (nout)

!     Generate an array of data

      Do i = 1, la
        a(i) = real(i,kind=nag_wp)
      End Do
```

```
!     Print order (nmax-1) lower triangular matrix

      ifail = 0
      Call x04ccf('Lower','Unit',nmax-1,a,'Example 1:',ifail)

      Write (nout,*)
      Flush (nout)

!     Print order nmax upper triangular matrix

      ifail = 0
      Call x04ccf('Upper','Non-unit',nmax,a,'Example 2:',ifail)

    End Program x04ccfe
```

9.2 Program Data

None.

9.3 Program Results

```
 X04CCF Example Program Results

 Example 1:
             1          2          3          4
 1      1.0000
 2      2.0000     1.0000
 3      3.0000     6.0000     1.0000
 4      4.0000     7.0000     9.0000     1.0000

 Example 2:
             1          2          3          4          5
 1      1.0000     2.0000     4.0000     7.0000    11.0000
 2                 3.0000     5.0000     8.0000    12.0000
 3                            6.0000     9.0000    13.0000
 4                                      10.0000    14.0000
 5                                                 15.0000
```