NAG Fortran Library Routine Document

X04DCF

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

X04DCF is an easy-to-use routine to print a ***complex*** triangular matrix stored in a packed one-dimensional array.

2 Specification

```
SUBROUTINE X04DCF(UPLO, DIAG, N, A, TITLE, IFAIL)
INTEGER           N, IFAIL
complex           A(*)
CHARACTER*1       UPLO, DIAG
CHARACTER*(*)     TITLE
```

3 Description

X04DCF prints a ***complex*** triangular matrix stored in packed form. It is an easy-to-use driver for X04DDF. 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.

X04DCF 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 chosen code is used to print each complex element of the matrix with the real part above the imaginary part.

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, as follows:

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

if UPLO = 'U' (Upper), 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 must be 'L' or 'U'.

2: DIAG – CHARACTER*1 *Input*

On entry: indicates whether the diagonal elements of the matrix are to be printed, as follows:

if DIAG = 'B' (Blank), the diagonal elements of the matrix are not referenced and not printed;

if DIAG = 'U' (Unit diagonal), the diagonal elements of the matrix are not referenced, but are assumed all to be unity, and are printed as such;

if DIAG = 'N' (Non-unit diagonal), the diagonal elements of the matrix are referenced and printed.

Constraint: DIAG must be one of 'B', 'U' or 'N'.

3: N – INTEGER *Input*

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

If N is less than 1, X04DCF will exit immediately after printing TITLE; no row or column labels are printed.

4: A(∗) – ***complex*** array *Input*

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

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.

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. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

On exit: 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, UPLO ≠ 'L' or 'U'.

IFAIL = 2

On entry, DIAG ≠ 'N', 'U' or 'B'.

7 Accuracy

Not applicable.

8 Further Comments

A call to X04DCF is equivalent to a call to X04DDF with the following argument values:

```
NCOLS = 80
INDENT = 0
LABROW = 'I'
LABCOL = 'I'
FORMAT = ' '
USEFRM = 'A'
```

9 Example

The example program calls X04DCF twice, first to print a 3 by 3 lower triangular matrix, and then to print a 4 by 4 upper triangular matrix.

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.

```
*     X04DCF Example Program Text
*     Mark 14 Release.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
      INTEGER          NMAX, LA
      PARAMETER        (NMAX=4,LA=(NMAX*(NMAX+1))/2)
*     .. Local Scalars ..
      real             AA
      INTEGER          I, IFAIL
*     .. Local Arrays ..
      complex          A(LA)
*     .. External Subroutines ..
      EXTERNAL         X04DCF
*     .. Intrinsic Functions ..
      INTRINSIC        cmplx
*     .. Executable Statements ..
      WRITE (NOUT,*) 'X04DCF Example Program Results'
      WRITE (NOUT,*)
*
*     Generate an array of data
      DO 20 I = 1, LA
         AA = I
         A(I) = cmplx(AA,-AA)
   20 CONTINUE
*
      IFAIL = 0
*
*     Print order 3 lower triangular matrix
      CALL X04DCF('Lower','Unit',3,A,'Example 1:',IFAIL)
*
      WRITE (NOUT,*)
*
*     Print order 4 upper triangular matrix
      CALL X04DCF('Upper','Non-unit',4,A,'Example 2:',IFAIL)
*
      STOP
      END
```

9.2 Program Data

None.

9.3 Program Results

```
 X04DCF Example Program Results

 Example 1:
              1          2          3
 1       1.0000
         0.0000

 2       2.0000     1.0000
        -2.0000     0.0000

 3       3.0000     5.0000     1.0000
        -3.0000    -5.0000     0.0000

 Example 2:
              1          2          3          4
 1       1.0000     2.0000     4.0000     7.0000
        -1.0000    -2.0000    -4.0000    -7.0000

 2                  3.0000     5.0000     8.0000
                   -3.0000    -5.0000    -8.0000

 3                             6.0000     9.0000
                              -6.0000    -9.0000

 4                                       10.0000
                                        -10.0000
```