

NAG Library Routine Document

X04DFF

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

X04DFF prints a ***complex*16*** band matrix stored in a packed two-dimensional array.

2 Specification

```

SUBROUTINE X04DFF(M, N, KL, KU, A, LDA, USEFRM, FORM, TITLE, LABROW,
1              RLABS, LABCOL, CLABS, NCOLS, INDENT, IFAIL)
      INTEGER          M, N, KL, KU, LDA, NCOLS, INDENT, IFAIL
      complex*16        A(LDA,*)
      CHARACTER*(*)    FORM, TITLE, RLABS(*), CLABS(*)
      CHARACTER*1       USEFRM, LABROW, LABCOL

```

3 Description

X04DFF prints a ***complex*16*** band matrix stored in a packed two-dimensional array, using a format specifier supplied by you. The matrix is output to the unit defined by X04ABF.

4 References

None.

5 Parameters

- | | | |
|----|-------------|--------------|
| 1: | M – INTEGER | <i>Input</i> |
| 2: | N – INTEGER | <i>Input</i> |

On entry: the number of rows and columns of the band matrix, respectively, to be printed.

If either M or N is less than 1, X04DFF will exit immediately after printing TITLE; no row or column labels are printed.

- | | | |
|----|--------------|--------------|
| 3: | KL – INTEGER | <i>Input</i> |
|----|--------------|--------------|

On entry: the number of subdiagonals of the band matrix *A*.

Constraint: $KL \geq 0$.

- | | | |
|----|--------------|--------------|
| 4: | KU – INTEGER | <i>Input</i> |
|----|--------------|--------------|

On entry: the number of superdiagonals of the band matrix *A*.

Constraint: $KU \geq 0$.

- | | | |
|----|---|--------------|
| 5: | A(LDA,*) – <i>complex*16</i> array | <i>Input</i> |
|----|---|--------------|

Note: the second dimension of the array A must be at least $\max(1, \min(M + KU, N))$.

On entry: the band matrix to be printed.

The matrix is stored in rows 1 to $k_l + k_u + 1$, more precisely, the element A_{ij} must be stored in

$$A(k_u + 1 + i - j, j) \quad \text{for } \max(1, j - k_u) \leq i \leq \min(m, j + k_l).$$

- 6: LDA – INTEGER Input

On entry: the first dimension of the array A as declared in the (sub)program from which X04DFF is called.

Constraint: $LDA \geq KL + KU + 1$.

- 7: USEFRM – CHARACTER*1 Input

On entry: indicates how the value of FORM is to be used to print matrix elements.

USEFRM = 'A'

The format code in FORM is assumed to contain a single real edit-descriptor which is to be used to print the real and imaginary parts of each **complex*16** number one above the other. Each row of the matrix is separated by a blank line, and any row labels are attached only to the real parts. This option means that about twice as many columns can be fitted into NCOLS characters than if any other USEFRM option is used. A typical value of FORM for this USEFRM option might be 'E13.4', '*' or ' '.

USEFRM = 'B'

The format code in FORM is assumed to contain a single edit-descriptor such as 'E13.4', '*' or ' ' which is used to print the real and imaginary parts of each **complex*16** number separated by a comma, and surrounded by brackets. Thus a matrix element printed with this USEFRM option might look like this: (12.345, -11.323).

USEFRM = 'D'

The format code in FORM is used unaltered to print a **complex*16** number. This USEFRM option allows you flexibility to specify exactly how the number is printed. With this option for USEFRM and a suitable value for FORM it is possible, for example, to print a **complex*16** number in the form (0.123 + 3.214i) or (0.123D-02, 0.234D-01). See Section 9 for an example illustrating this option.

Constraint: USEFRM = 'A', 'B' or 'D'.

- 8: FORM – CHARACTER*(*) Input

On entry: a valid Fortran format code. This may be any format code allowed on the system, whether it is standard Fortran or not. FORM is used in conjunction with parameter USEFRM, to print elements of the matrix A. It may or may not be enclosed in brackets. Examples of valid values for FORM are '(F11.4)', '1P,2E13.5'.

In addition, there are two special codes which force X04DFF to choose its own format code:

FORM = ' '

X04DFF 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 the real and imaginary parts 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 numbers to be printed lie between 0.001 and 9999.9999. Otherwise the 1PE13.4 code is chosen.

FORM = '*'

X04DFF will choose a format code such that numbers will be printed to as many significant digits as are necessary to distinguish between neighbouring machine numbers. Thus any two numbers that are stored with different internal representations should look different on output. Whether they do in fact look different will depend on the run-time library of the Fortran compiler in use.

FORM = 'MATLAB' followed by any of the above, e.g., FORM = 'MATLAB F11.4',
FORM = 'MATLAB *'

X04DFF will print the matrix such that it can be input into MATLAB. Elements of the matrix will be printed with format specified by the format code following 'MATLAB'. TITLE will be used as the name of the matrix.

More complicated values of FORM, to print a **complex*16** number in a desired form, may be used. See the description of parameter USEFRM for more details.

Constraint: the character length of FORM must be ≤ 80 .

- 9: TITLE – CHARACTER*(*) *Input*
On entry: a title to be printed above the matrix or name of the matrix.
 If TITLE = ' ', no title (and no blank line) will be printed.
 If TITLE contains more than NCOLS characters, the contents of TITLE will be wrapped onto more than one line, with the break after NCOLS characters.
 Any trailing blank characters in TITLE are ignored.
 If FORM = 'MATLAB', TITLE will be used as the name of the matrix.
- 10: LABROW – CHARACTER*1 *Input*
On entry: indicates the type of labelling to be applied to the rows of the matrix.
 LABROW = 'N'
 Prints no row labels.
 LABROW = 'I'
 Prints integer row labels.
 LABROW = 'C'
 Prints character labels, which must be supplied in array RLABS.
Constraint: LABROW = 'N', 'I' or 'C'.
- 11: RLABS(*) – CHARACTER*(*) array *Input*
Note: the dimension of the array RLABS must be at least M if LABROW = 'C', and at least 1 otherwise.
On entry: if LABROW = 'C', must contain labels for the rows of the matrix.
 Labels are right-justified when output, in a field which is as wide as necessary to hold the longest row label. Note that this field width is subtracted from the number of usable columns, NCOLS.
- 12: LABCOL – CHARACTER*1 *Input*
On entry: indicates the type of labelling to be applied to the columns of the matrix.
 LABCOL = 'N'
 Prints no column labels.
 LABCOL = 'I'
 Prints integer column labels.
 LABCOL = 'C'
 Prints character labels, which must be supplied in array CLABS.
Constraint: LABCOL = 'N', 'I' or 'C'.
- 13: CLABS(*) – CHARACTER*(*) array *Input*
Note: the dimension of the array CLABS must be at least N if LABCOL = 'C', and at least 1 otherwise.
On entry: if LABCOL = 'C', must contain labels for the columns of the matrix.
 Labels are right-justified when output. Any label that is too long for the column width, which is determined by FORM, is truncated.

- 14: NCOLS – INTEGER *Input*
- On entry:* the maximum output record length. If the number of columns of the matrix is too large to be accommodated in NCOLS characters, the matrix will be printed in parts, containing the largest possible number of matrix columns, and each part separated by a blank line.
- NCOLS must be large enough to hold at least one column of the matrix using the format specifier in FORM. If a value less than 0 or greater than 132 is supplied for NCOLS, then the value 80 is used instead.
- 15: INDENT – INTEGER *Input*
- On entry:* the number of columns by which the matrix (and any title and labels) should be indented. The effective value of NCOLS is reduced by INDENT columns. If a value less than 0 or greater than NCOLS is supplied for INDENT, the value 0 is used instead.
- 16: 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.
- 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, 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.**

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, $KL < 0$.

IFAIL = 2

On entry, $KU < 0$.

IFAIL = 3

On entry, $LDA < KL + KU + 1$.

IFAIL = 4

On entry, USEFRM $\neq$ 'A', 'B' or 'D'.

IFAIL = 5

On entry, variable FORM is more than 80 characters long.

IFAIL = 6

The code supplied in FORM cannot be used to output a number. FORM probably has too wide a field width or contains an illegal edit descriptor.

IFAIL = 7

On entry, either LABROW or LABCOL $\neq$ 'N', 'I' or 'C'.

IFAIL = 8

The quantity $\text{NCOLS} - \text{INDENT} - \text{labwid}$ (where *labwid* is the width needed for the row labels) is not large enough to hold at least one column of the matrix.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

The example program calls X04DFF three times, to print band matrices of different orders and bandwidths; various options for labelling and formatting are illustrated.

9.1 Program Text

```
*      X04DFF Example Program Text
*      Mark 14 Release. NAG Copyright 1989.
*      .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
      INTEGER          NMAX, LDA
      PARAMETER        (NMAX=5,LDA=NMAX)
*      .. Local Scalars ..
      DOUBLE PRECISION AA
      INTEGER          I, IFAIL, INDENT, J, NCOLS
      CHARACTER*19     FORM
*      .. Local Arrays ..
      COMPLEX *16      A(LDA,LDA)
      CHARACTER*7       CLABS(NMAX), RLABS(NMAX)
*      .. External Subroutines ..
      EXTERNAL         X04DFF
*      .. Intrinsic Functions ..
      INTRINSIC        CMPLX
*      .. Data statements ..
      DATA             CLABS/'Un', 'Deux', 'Trois', 'Quatre', 'Cinq'/
      DATA             RLABS/'Uno', 'Duo', 'Tre', 'Quattro', 'Cinque'/
*      .. Executable Statements ..
      WRITE (NOUT,*) 'X04DFF Example Program Results'
      WRITE (NOUT,*)

*
*      Generate an array of data
      DO 40 J = 1, NMAX
        DO 20 I = 1, LDA
          AA = 10*I + J
          A(I,J) = CMPLX(AA,-AA,KIND=KIND(AA))
20      CONTINUE
40     CONTINUE
*
      NCOLS = 80
      INDENT = 0
      IFAIL = 0
      FORM = ' '
*
*      Print 5 by 5 band matrix with 1 sub-diagonal, 1 super-diagonal,
*      default format, bracketed complex numbers, and integer row and
*      column labels
      CALL X04DFF(5,5,1,1,A,LDA,'Bracketed',FORM,'Example 1:', 'Integer',
+           RLABS, 'Integer', CLABS, NCOLS, INDENT, IFAIL)
*
      WRITE (NOUT,*)
      FORM = 'SS,F7.1,SP,F6.1,''i'''
```

```

*
*   Print 4 by 4 band matrix with 1 sub-diagonal, 2 super-diagonals,
*   user-supplied format and row and column labels
*   CALL X04DFF(4,4,1,2,A,LDA,'Direct',FORM,'Example 2:', 'Character',
+             RLABS,'Character',CLABS,NCOLS,INDENT,IFAIL)
*
      END

```

9.2 Program Data

None.

9.3 Program Results

X04DFF Example Program Results

Example 1:

```

              1              2
1 ( 21.0000, -21.0000) ( 12.0000, -12.0000)
2 ( 31.0000, -31.0000) ( 22.0000, -22.0000)
3              ( 32.0000, -32.0000)
4
5

```

```

              3              4
1
2 ( 13.0000, -13.0000)
3 ( 23.0000, -23.0000) ( 14.0000, -14.0000)
4 ( 33.0000, -33.0000) ( 24.0000, -24.0000)
5              ( 34.0000, -34.0000)

```

```

              5
1
2
3
4 ( 15.0000, -15.0000)
5 ( 25.0000, -25.0000)

```

Example 2:

	Un	Deux	Trois	Quatre
Uno	31.0 -31.0i	22.0 -22.0i	13.0 -13.0i	
Duo	41.0 -41.0i	32.0 -32.0i	23.0 -23.0i	14.0 -14.0i
Tre		42.0 -42.0i	33.0 -33.0i	24.0 -24.0i
Quattro			43.0 -43.0i	34.0 -34.0i
