

NAG Fortran Library Routine Document

X04EBF

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

X04EBF prints an integer matrix stored in a two-dimensional array.

2 Specification

```

SUBROUTINE X04EBF(MATRIX, DIAG, M, N, A, LDA, FORMAT, TITLE, LABROW,
1              RLABS, LABCOL, CLABS, NCOLS, INDENT, IFAIL)
INTEGER          M, N, A(LDA,*), LDA, NCOLS, INDENT, IFAIL
CHARACTER*1      MATRIX, DIAG, LABROW, LABCOL
CHARACTER*(*)    FORMAT, TITLE, RLABS(*), CLABS(*)

```

3 Description

X04EBF prints an integer matrix, or part of it, using a format specifier supplied by the user. The matrix is output to the unit defined by X04ABF.

4 References

None.

5 Parameters

1: MATRIX – CHARACTER*1 *Input*

On entry: indicates the part of the matrix to be printed, as follows:

if MATRIX = 'G' (General), the whole of the rectangular matrix;

if MATRIX = 'L' (Lower), the lower triangle of the matrix, or the lower trapezium if the matrix has more rows than columns;

if MATRIX = 'U' (Upper), the upper triangle of the matrix, or the upper trapezium if the matrix has more columns than rows.

Constraint: MATRIX must be one of 'G', 'L', 'U'.

2: DIAG – CHARACTER*1 *Input*

On entry: unless MATRIX = 'G', DIAG must specify 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.

If MATRIX = 'G', then DIAG need not be set.

Constraint: If MATRIX ≠ 'G', then DIAG must be one of 'B', 'U' or 'N'.

- 3: M – INTEGER *Input*
- 4: N – INTEGER *Input*
- On entry:* the number of rows and columns of the matrix, respectively, to be printed.
- If either M or N is less than 1, X04EBF will exit immediately after printing TITLE; no row or column labels are printed.
- 5: A(LDA,*) – INTEGER array *Input*
- Note:** the second dimension of the array A must be at least $\max(1, N)$.
- On entry:* the matrix to be printed. Only the elements that will be referred to, as specified by parameters MATRIX and DIAG, need be set.
- 6: LDA – INTEGER *Input*
- On entry:* the first dimension of the array A as declared in the (sub)program from which X04EBF is called.
- Constraint:* $LDA \geq M$.
- 7: FORMAT – 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. FORMAT is used to print elements of the matrix A. It may or may not be enclosed in brackets. Examples of valid values for FORMAT are 'I6', 'I4,2X'.
- In addition, there is a special code which forces X04EBF to choose its own format code:
- if $FORMAT = ' '$, X04EBF will choose a format code such that numbers will be printed using the smallest edit descriptor that is large enough to hold all the numbers to be printed.
- Constraint:* the character length of FORMAT must be ≤ 80 .
- 8: 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 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.
- 9: LABROW – CHARACTER*1 *Input*
- On entry:* indicates the type of labelling to be applied to the rows of the matrix, as follows:
- if $LABROW = 'N'$, X04EBF prints no row labels;
- if $LABROW = 'I'$, X04EBF prints integer row labels;
- if $LABROW = 'C'$, X04EBF prints character labels, which must be supplied in array RLABS.
- Constraint:* $LABROW = 'N', 'I' \text{ or } 'C'$.
- 10: RLABS(*) – CHARACTER*(*) array *Input*
- On entry:* if $LABROW = 'C'$, RLABS must be dimensioned at least of length M and must contain labels for the rows of the matrix, otherwise RLABS may be dimensioned of length 1.
- 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.

- 11: LABCOL – CHARACTER*1 *Input*
On entry: indicates the type of labelling to be applied to the columns of the matrix, as follows:
 if LABCOL = 'N', X04EBF prints no column labels;
 if LABCOL = 'I', X04EBF prints integer column labels;
 if LABCOL = 'C', X04EBF prints character labels, which must be supplied in array CLABS.
Constraint: LABCOL = 'N', 'I' or 'C'.
- 12: CLABS(*) – CHARACTER*(*) array *Input*
On entry: if LABCOL = 'C', CLABS must be dimensioned at least of length N and must contain labels for the columns of the matrix, otherwise CLABS may be dimensioned of length 1.
 Labels are right-justified when output. Any label that is too long for the column width, which is determined by FORMAT, is truncated.
- 13: 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 FORMAT. If a value less than 0 or greater than 132 is supplied for NCOLS, then the value 80 is used instead.
- 14: 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.
- 15: 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, MATRIX \neq 'G', 'L' or 'U'.

IFAIL = 2

On entry, MATRIX = 'L' or 'U', but DIAG \neq 'N', 'U' or 'B'.

IFAIL = 3

On entry, M > LDA.

IFAIL = 4

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

IFAIL = 5

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

IFAIL = 6

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

IFAIL = 7

The quantity NCOLS – INDENT – 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

X04EBF may be used to print a vector, either as a row or as a column. The following code fragment illustrates possible calls.

```

      INTEGER A(4)
      CHARACTER*1 RLABS(1), CLABS(1)
C     Print vector A as a column vector.
      LDA = 4
      IFAIL = 0
      CALL X04EBF('G','X',1,4,A,LDA,' ',' ','I',RLABS,
*              'N',CLABS,80,0,IFAIL)
C     Print vector A as a row vector.
      LDA = 1
      IFAIL = 0
      CALL X04EBF('G','X',4,1,A,LDA,' ',' ','N',RLABS,
*              'I',CLABS,80,0,IFAIL)

```

9 Example

The example program calls X04EBF twice, first to print a 3 by 5 rectangular matrix, and then to print a 5 by 5 upper triangular matrix; various options for labelling and formatting are illustrated.

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.

```

*     X04EBF 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 ..
      INTEGER          I, IFAIL, INDENT, J, NCOLS
*     .. Local Arrays ..
      INTEGER          A(LDA,NMAX)
      CHARACTER*7      CLABS(NMAX), RLABS(NMAX)
*     .. External Subroutines ..
      EXTERNAL         X04EBF

```

```

*      .. Data statements ..
DATA          CLABS/'Un', 'Deux', 'Trois', 'Quatre', 'Cinq'/
DATA          RLABS/'Uno', 'Duo', 'Tre', 'Quattro', 'Cinque'/
*      .. Executable Statements ..
WRITE (NOUT,*) 'X04EBF Example Program Results'
WRITE (NOUT,*)
*      Generate an array of data
DO 40 J = 1, NMAX
    DO 20 I = 1, LDA
        A(I,J) = 10*I + J
20    CONTINUE
40 CONTINUE
    NCOLS = 80
    INDENT = 0
    IFAIL = 0
*
*      Print 3 by 5 rectangular matrix with default format and integer
*      row and column labels
CALL X04EBF('General', ' ', 3, 5, A, LDA, ' ', 'Example 1:', 'Integer',
+          RLABS, 'Integer', CLABS, NCOLS, INDENT, IFAIL)
*
WRITE (NOUT,*)
*
*      Print 5 by 5 upper triangular matrix with user-supplied format
*      and row and column labels
CALL X04EBF('Upper', 'Non-unit', 5, 5, A, LDA, 'I8', 'Example 2:',
+          'Character', RLABS, 'Character', CLABS, NCOLS, INDENT,
+          IFAIL)
*
STOP
END

```

9.2 Program Data

None.

9.3 Program Results

X04EBF Example Program Results

Example 1:

```

  1  2  3  4  5
1  11 12 13 14 15
2  21 22 23 24 25
3  31 32 33 34 35

```

Example 2:

	Un	Deux	Trois	Quatre	Cinq
Uno	11	12	13	14	15
Duo		22	23	24	25
Tre			33	34	35
Quattro				44	45
Cinque					55
