

NAG 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, FORM, TITLE, LABROW,      &
                  RLABS, LABCOL, CLABS, NCOLS, INDENT, IFAIL)
INTEGER          M, N, A(LDA,*), LDA, NCOLS, INDENT, IFAIL
CHARACTER(*)    FORM, TITLE, RLABS(*), CLABS(*)
CHARACTER(1)    MATRIX, DIAG, LABROW, LABCOL
```

3 Description

X04EBF prints an integer matrix, or part of it, using a format specifier supplied by you. 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.
 MATRIX = 'G'
 The whole of the rectangular matrix.
 MATRIX = 'L'
 The lower triangle of the matrix, or the lower trapezium if the matrix has more rows than columns.
 MATRIX = 'U'
 The upper triangle of the matrix, or the upper trapezium if the matrix has more columns than rows.
Constraint: MATRIX = 'G', 'L' or '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.
 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.

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

Constraint: if MATRIX \neq 'G', DIAG = '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: $LDA \geq \max(1, M)$.

7: 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 to print elements of the matrix A. It may or may not be enclosed in brackets. Examples of valid values for FORM are 'I6', 'I4,2X'.

In addition, there is a special code which forces X04EBF to choose its own format code:

FORM = ' '

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.

FORM = 'MATLAB' followed by any of the above, e.g., FORM = 'MATLABI6', FORM = 'MATLAB*'

X04EBF 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.

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

8: 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 printing in MATLAB mode, TITLE will be used as the name of the matrix.

9: LABROW – CHARACTER(1) *Input*

On entry: indicates the type of labelling to be applied to the rows of the matrix, except in MATLAB mode where LABROW is ignored.

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'.

- 10: 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', RLABS must contain labels for the rows of the matrix, except in MATLAB mode where RLABS is ignored.

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, except in MATLAB mode where LABCOL is ignored.

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'.

- 12: 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', CLABS must contain labels for the columns of the matrix, except in MATLAB mode where CLABS is ignored.

Labels are right-justified when output. Any label that is too long for the column width, which is determined by FORM, 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 FORM. 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. 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, 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, the format specifier in FORM is more than 80 characters long.

IFAIL = 5

The format specifier in FORM cannot be used to output a number. The specifier 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.

IFAIL = -99

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.8 in the Essential Introduction for further information.

IFAIL = -399

Your licence key may have expired or may not have been installed correctly.

See Section 3.7 in the Essential Introduction for further information.

IFAIL = -999

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

7 Accuracy

Not applicable.

8 Parallelism and Performance

Not applicable.

9 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)
!      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)
!      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)

```

10 Example

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

10.1 Program Text

```

Program x04ebfe

!      X04EBF Example Program Text

!      Mark 25 Release. NAG Copyright 2014.

!      .. Use Statements ..
      Use nag_library, Only: x04ebf
!      .. Implicit None Statement ..
      Implicit None
!      .. Parameters ..
      Integer, Parameter          :: nmax = 5, nout = 6
      Integer, Parameter          :: lda = nmax
      Character (7), Parameter    :: clabs(nmax) = (/
          'Un      ','Deux  ','
          'Trois  ','Quatre ','Cinq   ' /)
      Character (7), Parameter    :: rlabs(nmax) = (/
          'Uno     ','Due   ','
          'Tre    ','Quattro','Cinque ' /)

!      .. Local Scalars ..
      Integer                      :: i, ifail, indent, j, ncols
!      .. Local Arrays ..
      Integer                      :: a(lda,nmax)
!      .. Executable Statements ..
      Write (nout,*) 'X04EBF Example Program Results'

      Write (nout,*)
      Flush (nout)

!      Generate an array of data

      Do j = 1, nmax

```

```

      Do i = 1, lda
        a(i,j) = 10*i + j
      End Do

End Do

ncols = 80
indent = 0

!   Print 3 by nmax rectangular matrix with default format and integer
!   row and column labels

ifail = 0
Call x04ebf('General',' ',3,nmax,a,lda,' ','Example 1:','Integer',rlabs, &
  'Integer',clabs,ncols,indent,ifail)

Write (nout,*)
Flush (nout)

!   Print nmax by nmax upper triangular matrix with user-supplied format
!   and row and column labels

ifail = 0
Call x04ebf('Upper','Non-unit',nmax,nmax,a,lda,'I8','Example 2:', &
  'Character',rlabs,'Character',clabs,ncols,indent,ifail)

Write (nout,*)
Flush (nout)

!   Print 4 by nmax lower triangular matrix in MATLAB format
!   Row and column labelling is ignored

ifail = 0
Call x04ebf('Lower','Non-unit',4,nmax,a,lda,'MATLABI8','A',' ',rlabs, &
  ' ',clabs,ncols,indent,ifail)

End Program x04ebfe

```

10.2 Program Data

None.

10.3 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
Due		22	23	24	25
Tre			33	34	35
Quattro				44	45
Cinque					55

```

A = [
  11    0    0    0    0;
  21    22    0    0    0;
  31    32    33    0    0;
  41    42    43    44    0;
];

```
