

# NAG Library Routine Document

## X04EAF

**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

X04EAF is an easy-to-use routine to print an integer matrix stored in a two-dimensional array.

### 2 Specification

```
SUBROUTINE X04EAF (MATRIX, DIAG, M, N, A, LDA, TITLE, IFAIL)
INTEGER          M, N, A(LDA,*), LDA, IFAIL
CHARACTER(*)    TITLE
CHARACTER(1)    MATRIX, DIAG
```

### 3 Description

X04EAF prints an integer matrix. It is an easy-to-use driver for X04EBF. The routine uses default values for the format in which numbers are printed, for labelling the rows and columns, and for output record length.

X04EAF will choose a format code such that numbers will be printed with the smallest I edit descriptor that is large enough to hold all the numbers to be printed.

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 Arguments

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, X04EAF 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 arguments 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 X04EAF is called.

*Constraint:*  $LDA \geq \max(1, M)$ .

7: 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.

8: IFAIL – INTEGER *Input/Output*

*On entry:* IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation 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 argument, 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, `LDA < M`.

`IFAIL = -99`

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

See Section 3.9 in *How to Use the NAG Library and its Documentation* for further information.

`IFAIL = -399`

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

See Section 3.8 in *How to Use the NAG Library and its Documentation* for further information.

`IFAIL = -999`

Dynamic memory allocation failed.

See Section 3.7 in *How to Use the NAG Library and its Documentation* for further information.

## 7 Accuracy

Not applicable.

## 8 Parallelism and Performance

X04EAF is not threaded in any implementation.

## 9 Further Comments

A call to X04EAF is equivalent to a call to X04EBF with the following argument values:

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

## 10 Example

This example calls X04EAF twice, first to print a 3 by 5 rectangular matrix, and then to print a 5 by 5 triangular matrix.

## 10.1 Program Text

```

Program x04eafe

!      X04EAF Example Program Text

!      Mark 26 Release. NAG Copyright 2016.

!      .. Use Statements ..
      Use nag_library, Only: x04eaf
!      .. Implicit None Statement ..
      Implicit None
!      .. Parameters ..
      Integer, Parameter          :: nmax = 5, nout = 6
      Integer, Parameter          :: lda = nmax
!      .. Local Scalars ..
      Integer                      :: i, ifail, j
!      .. Local Arrays ..
      Integer                      :: a(lda,nmax)
!      .. Executable Statements ..
      Write (nout,*) 'X04EAF 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

!      Print 3 by nmax rectangular matrix

      ifail = 0
      Call x04eaf('General', ' ', 3, nmax, a, lda, 'Example 1:', ifail)

      Write (nout,*)
      Flush (nout)

!      Print nmax by nmax lower triangular matrix

      ifail = 0
      Call x04eaf('Lower', 'Non-unit', nmax, nmax, a, lda, 'Example 2:', ifail)

End Program x04eafe

```

## 10.2 Program Data

None.

## 10.3 Program Results

X04EAF 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:

```

  1  2  3  4  5

```

1 11  
2 21 22  
3 31 32 33  
4 41 42 43 44  
5 51 52 53 54 55

---