

NAG Library Routine Document

X04DEF

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

X04DEF is an easy-to-use routine to print a complex band matrix stored in a packed two-dimensional array.

2 Specification

```
SUBROUTINE X04DEF (M, N, KL, KU, A, LDA, TITLE, IFAIL)
  INTEGER          M, N, KL, KU, LDA, IFAIL
  COMPLEX (KIND=nag_wp) A(LDA,*)
  CHARACTER(*)     TITLE
```

3 Description

X04DEF prints a complex band matrix stored in a packed two-dimensional array. It is an easy-to-use driver for X04DFF. The routine uses default values for the format in which numbers are printed, for labelling the rows and columns, and for output record length.

X04DEF 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 Arguments

- | | | |
|----|-------------|--------------|
| 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, X04DEF 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,*) – COMPLEX (KIND=nag_wp) array *Input*
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 X04DEF is called.
Constraint: $LDA \geq KL + KU + 1$.
- 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, $KL < 0$.

IFAIL = 2

On entry, $KU < 0$.

IFAIL = 3

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

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

X04DEF is not threaded in any implementation.

9 Further Comments

A call to X04DEF is equivalent to a call to X04DEF with the following argument values:

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

10 Example

This example program calls X04DEF to print a 5 by 5 band matrix with one subdiagonal and one superdiagonal.

10.1 Program Text

```
Program x04defe

!      X04DEF Example Program Text

!      Mark 26 Release. NAG Copyright 2016.

!      .. Use Statements ..
!      Use nag_library, Only: f01zdf, nag_wp, x04def
!      .. Implicit None Statement ..
!      Implicit None
!      .. Parameters ..
!      Integer, Parameter          :: nout = 6
!      .. Local Scalars ..
!      Real (Kind=nag_wp)         :: aa
!      Integer                    :: i, ifail, j, kl, ku, lda, ldab, m, n
!      Character (1)              :: job
!      .. Local Arrays ..
!      Complex (Kind=nag_wp), Allocatable :: a(:,,:), ab(:,,:)
!      .. Intrinsic Procedures ..
!      Intrinsic                   :: cmlpx, max, min, real
!      .. Executable Statements ..
!      Write (nout,*) 'X04DEF Example Program Results'

!      Write (nout,*)
!      Flush (nout)

!      Generate a square array of data.
```

```

      m = 5
      n = m
      kl = 1
      ku = 1

      lda = m
      Allocate (a(lda,n))

      Do j = 1, n

         Do i = max(1,j-ku), min(m,j+kl)
            aa = real(10*i+j,kind=nag_wp)
            a(i,j) = cmplx(aa,-aa,kind=nag_wp)
         End Do

      End Do

!      Convert to packed storage.

      ldab = kl + ku + 1
      Allocate (ab(ldab,n))

      job = 'P'

      ifail = 0
      Call f01zdf(job,m,n,kl,ku,a,lda,ab,ldab,ifail)

!      Print m by n band matrix with kl subdiagonals and ku superdiagonals.

      ifail = 0
      Call x04def(m,n,kl,ku,ab,ldab,'Band Matrix:',ifail)

End Program x04defe

```

10.2 Program Data

None.

10.3 Program Results

X04DEF Example Program Results

Band Matrix:					
	1	2	3	4	5
1	11.0000	12.0000			
	-11.0000	-12.0000			
2	21.0000	22.0000	23.0000		
	-21.0000	-22.0000	-23.0000		
3		32.0000	33.0000	34.0000	
		-32.0000	-33.0000	-34.0000	
4			43.0000	44.0000	45.0000
			-43.0000	-44.0000	-45.0000
5				54.0000	55.0000
				-54.0000	-55.0000
