NAG Library Function Document

nag_band_complx_mat_print (x04dec)

1 Purpose

nag_band_complx_mat_print (x04dec) is an easy-to-use function to print a Complex band matrix.

2 Specification

```
#include <nag.h>
#include <nagx04.h>
```

```
void nag_band_complx_mat_print (Nag_OrderType order, Integer m, Integer n,
     Integer kl, Integer ku, const Complex a[], Integer pda,
     const char *title, const char *outfile, NagError *fail)
```

3 Description

nag_band_complx_mat_print (x04dec) prints a Complex band matrix stored in packed form. It is an easy-to-use driver for nag_band_complx_mat_print_comp (x04dfc). The function uses default values for the format in which numbers are printed, for labelling the rows and columns, and for output record length.

nag_band_complx_mat_print (x04dec) will choose a format code such that numbers will be printed with a %8.4f, a %11.4f or a %13.4e format. The %8.4f code is chosen if the sizes of all the matrix elements to be printed lie between 0.001 and 1.0. The %11.4f code is chosen if the sizes of all the matrix elements to be printed lie between 0.001 and 9999.9999. Otherwise the %13.4e 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 file specified by **outfile** or, by default, to standard output.

4 References

None.

5 Arguments

1: **order** – Nag_OrderType *Input*

On entry: the **order** argument specifies the two-dimensional storage scheme being used, i.e., row-major ordering or column-major ordering. C language defined storage is specified by **order** = Nag_RowMajor. See Section 3.2.1.3 in the Essential Introduction for a more detailed explanation of the use of this argument.

Constraint: **order** = Nag_RowMajor or Nag_ColMajor.

2: **m** – Integer *Input*

3: **n** – Integer *Input*

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, nag_band_complx_mat_print (x04dec) will exit immediately after printing **title**; no row or column labels are printed.

4: **kl** – Integer *Input*

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

Constraint: $\mathbf{kl} \geq 0$.

5: **ku** – Integer *Input*

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

Constraint: $\mathbf{ku} \geq 0$.

6: $\mathbf{a}[\mathit{dim}]$ – const Complex *Input*

Note: the dimension, *dim*, of the array **a** must be at least

$\max(1, \mathbf{pda} \times \mathbf{n})$ when **order** = Nag_ColMajor;
$\max(1, \mathbf{m} \times \mathbf{pda})$ when **order** = Nag_RowMajor.

On entry: the band matrix to be printed.

This is stored as a notional two-dimensional array with row elements or column elements stored contiguously. The storage of elements A_{ij}, for row $i = 1, \dots, m$ and column $j = \max(1, i - k_l), \dots, \min(n, i + k_u)$, depends on the **order** argument as follows:

if **order** = Nag_ColMajor, A_{ij} is stored as $\mathbf{a}[(j-1) \times \mathbf{pda} + \mathbf{ku} + i - j]$;

if **order** = Nag_RowMajor, A_{ij} is stored as $\mathbf{a}[(i-1) \times \mathbf{pda} + \mathbf{kl} + j - i]$.

7: **pda** – Integer *Input*

On entry: the stride separating row or column elements (depending on the value of **order**) of the matrix A in the array **a**.

Constraint: $\mathbf{pda} \geq \mathbf{kl} + \mathbf{ku} + 1$.

8: **title** – const char * *Input*

On entry: a title to be printed above the matrix.

If **title** = **NULL**, 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.

9: **outfile** – const char * *Input*

On entry: the name of a file to which output will be directed. If **outfile** is **NULL** the output will be directed to standard output.

10: **fail** – NagError * *Input/Output*

The NAG error argument (see Section 3.6 in the Essential Introduction).

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Memory allocation failed.

NE_BAD_PARAM

On entry, argument ⟨*value*⟩ had an illegal value.

NE_INTERNAL_ERROR

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please contact NAG for assistance.

NE_NOT_APPEND_FILE

Cannot open file ⟨*value*⟩ for appending.

NE_NOT_CLOSE_FILE

Cannot close file ⟨*value*⟩.

NE_NOT_WRITE_FILE

Cannot open file ⟨*value*⟩ for writing.

7 Accuracy

Not applicable.

8 Parallelism and Performance

Not applicable.

9 Further Comments

A call to nag_band_complx_mat_print (x04dec) is equivalent to a call to nag_band_complx_mat_print_comp (x04dfc) with the following argument values:

```
ncols = 80
indent = 0
labrow = Nag_IntegerLabels
labcol = Nag_IntegerLabels
form = 0
cmplxform = Nag_AboveForm
```

10 Example

None.