NAG C Library Function Document
nag_band_real_mat_print_comp (x04cfc)

1 Purpose

nag_band_real_mat_print_comp (x04cfc) prints a real band matrix.

2 Specification

void nag_band_real_mat_print_comp (Nag_OrderType order, Integer m, Integer n, Integer kl, Integer ku, const double a[], Integer pda, const char *format, const char *title, Nag_LabelType labrow, const char *rlabs[], Nag_LabelType labcol, const char *clabs[], Integer ncols, Integer indent, const char *outfile, NagError *fail)

3 Description

nag_band_real_mat_print_comp (x04cfc) prints a real band matrix stored in packed form, using a format specifier supplied by the user. The matrix is output to the file specified by outfile or, by default, to standard output.

4 References

None.

5 Parameters

1: order – Nag_OrderType

On entry: the order parameter 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 2.2.1.4 of the Essential Introduction for a more detailed explanation of the use of this parameter.

Constraint: order = Nag_RowMajor or Nag_ColMajor.

2: m – Integer

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_real_mat_print_comp (x04cfc) will exit immediately after printing title; no row or column labels are printed.

3: n – Integer

On entry: the number of rows and columns of the band matrix, respectively, to be printed.

Constraint: order = Nag_RowMajor or Nag_ColMajor.

4: kl – Integer

On entry: the number of sub-diagonals of the band matrix A.

Constraint: kl ≥ 0.

5: ku – Integer

On entry: the number of super-diagonals of the band matrix A.

Constraint: ku ≥ 0.

6: a[dim] – const double

Note: the dimension, dim, of the array a must be at least max(1,pda × n) when order = Nag_ColMajor and at least max(1,pda × m) when order = Nag_RowMajor.
On entry: the \( m \) by \( n \) matrix \( A \). This is stored as a notional two-dimensional array with row elements or column elements stored contiguously. The storage of elements \( a_{ij} \), for \( i = 1, \ldots, m \) and \( j = \max(1, i - k_l), \ldots, \min(n, i + k_u) \), depends on the \texttt{order} parameter as follows:

\[
\begin{align*}
\text{if } \texttt{order} &= \texttt{Nag-ColMajor}, & a_{ij} \text{ is stored as } a[j \times pda + kl + ku + i - j]; \\
\text{if } \texttt{order} &= \texttt{Nag-RowMajor}, & a_{ij} \text{ is stored as } a[(i - 1) \times pda + kl + j - i].
\end{align*}
\]

\textbf{7:} \texttt{pda} – Integer \hspace{1cm} Input

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

\textbf{Constraint:} \texttt{pda} \geq kl + ku + 1.

\textbf{8:} \texttt{format} – char * \hspace{1cm} Input

On entry: a valid C format code. This should be of the form \( %[flag]ww.pp[format indicator] \), where \( ww.pp \) indicates that up to 2 digits may be used to specify the field width and precision respectively. Only \% and \texttt{format indicator} must be present. \texttt{flag} can be one of \(-, +, < \text{ space } > \) or \# and \texttt{format indicator} can be e, E, f, g or G. Thus, possible formats include %f, %+23.15G, %.6e. \texttt{format} is used to print elements of the matrix \( A \).

In addition, nag\_band\_real\_mat\_print\_comp (x04cfc) chooses its own format code when \texttt{format} is \texttt{NULL} or \texttt{format} = *.

If \texttt{format} = \texttt{NULL}, nag\_band\_real\_mat\_print\_comp (x04cfc) will choose a format code such that numbers will be printed with either 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.

If \texttt{format} = *, nag\_band\_real\_mat\_print\_comp (x04cfc) will choose a format code such that numbers will be printed to as many significant digits as are necessary to distinguish between neighbouring machine numbers. Thus any two numbers that are stored with different internal representations should look different on output.

\textbf{Constraint:} \texttt{format} must be of the form \( %[flag]ww.pp[format indicator] \).

\textbf{9:} \texttt{title} – char * \hspace{1cm} Input

On entry: a title to be printed above the matrix. If \texttt{title} = \texttt{NULL}, no title (and no blank line) will be printed.

If \texttt{title} contains more than \texttt{ncols} characters, the contents of \texttt{title} will be wrapped onto more than one line, with the break after \texttt{ncols} characters.

Any trailing blank characters in \texttt{title} are ignored.

\textbf{10:} \texttt{labrow} – Nag\_LabelType \hspace{1cm} Input

On entry: indicates the type of labelling to be applied to the rows of the matrix, as follows:

\[
\begin{align*}
\text{if } \texttt{labrow} &= \texttt{Nag-NoLabels}, & \text{nag\_band\_real\_mat\_print\_comp (x04cfc) prints no row labels;} \\
\text{if } \texttt{labrow} &= \texttt{Nag-IntegerLabels}, & \text{nag\_band\_real\_mat\_print\_comp (x04cfc) prints integer row labels;} \\
\text{if } \texttt{labrow} &= \texttt{Nag-CharacterLabels}, & \text{nag\_band\_real\_mat\_print\_comp (x04cfc) prints character labels, which must be supplied in array \texttt{rlabs}.}
\end{align*}
\]

\textbf{Constraint:} \texttt{labrow} = \texttt{Nag-NoLabels}, \texttt{Nag-IntegerLabels} or \texttt{Nag-CharacterLabels}.

\textbf{11:} \texttt{rlabs[dim]} – const char * \hspace{1cm} Input

On entry: if \texttt{labrow} = \texttt{Nag-CharacterLabels}, \texttt{rlabs} must be dimensioned at least of length \( m \) and must contain labels for the rows of the matrix, otherwise \texttt{rlabs} should be \texttt{NULL}.
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.

12:  labelcol – Nag_LabelType  

*Input*

On entry: indicates the type of labelling to be applied to the columns of the matrix, as follows:

- if labelcol = Nag_no_labels, nag_band_real_mat_print_comp (x04cfc) prints no column labels;
- if labelcol = Nag_integer_labels, nag_band_real_mat_print_comp (x04cfc) prints integer column labels;
- if labelcol = Nag_character_labels, nag_band_real_mat_print_comp (x04cfc) prints character labels, which must be supplied in array clabs.

Constraint: labelcol = Nag_no_labels or Nag_integer_labels, or C.

13:  clabs[dim] – const char *  

*Input*

On entry: if labelcol = Nag_character_labels, clabs must be dimensioned at least of length n and must contain labels for the columns of the matrix, otherwise clabs should be NULL.

Labels are right-justified when output. Any label that is too long for the column width, which is determined by format, is truncated.

14:  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 or equal to 0 or greater than 132 is supplied for ncols, then the value 80 is used instead.

15:  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.

16:  outfile – 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.

17:  fail – NagError *  

*Input/Output*

The NAG error parameter (see the Essential Introduction).

6  Error Indicators and Warnings

**NE_COL_WIDTH**

(value) is not wide enough to hold at least one matrix column. ncols = (value), indent = (value).

**NE_INVALID_FORMAT**

The string (value), has not been recognised as a valid format.

**NE_ALLOC_FAIL**

Memory allocation failed.
NE_BAD_PARAM
On entry, parameter \(\text{value}\) had an illegal value.

NE_NOT_WRITE_FILE
Cannot open file \(\text{value}\) for writing.

NE_NOT_APPEND_FILE
Cannot open file \(\text{value}\) for appending.

NE_NOT_CLOSE_FILE
Cannot close file \(\text{value}\).

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 consult NAG for assistance.

7 Accuracy
Not applicable.

8 Further Comments
None.

9 Example
None.