NAG Toolbox

nag_file_print_matrix_real_gen_comp (x04cb)

1 Purpose

nag_file_print_matrix_real_gen_comp (x04cb) prints a real matrix stored in a two-dimensional array.

2 Syntax

```
[ifail] = nag_file_print_matrix_real_gen_comp(matrix, diag, a, form, title,
labrow, rlabs, labcol, clabs, ncols, indent, 'm', m, 'n', n)
```

```
[ifail] = x04cb(matrix, diag, a, form, title, labrow, rlabs, labcol, clabs,
ncols, indent, 'm', m, 'n', n)
```

3 Description

nag_file_print_matrix_real_gen_comp (x04cb) prints a double matrix, or part of it, using a format specifier supplied by you. The matrix is output to the unit defined by nag_file_set_unit_advisory (x04ab).

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **matrix** – CHARACTER(1)

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)

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: **a**$(lda,:)$ – REAL (KIND=nag_wp) array

The first dimension of the array **a** must be at least $\max(1,\mathbf{m})$.

The second dimension of the array **a** must be at least $\max(1,\mathbf{n})$.

The matrix to be printed. Only the elements that will be referred to, as specified by arguments **matrix** and **diag**, need be set.

4: **form** – CHARACTER(*)

Describes the Fortran format code for printing the elements of the matrix **a**. The format code may be any allowed on the system, whether it is standard Fortran or not. It may or may not be enclosed in brackets.

In addition, there are the following special codes which force nag_file_print_matrix_real_gen_comp (x04cb) to choose its own format code:

form $=$ ' '

nag_file_print_matrix_real_gen_comp (x04cb) 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.

form $=$ ' $*$ '

nag_file_print_matrix_real_gen_comp (x04cb) 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. Whether they do in fact look different will depend on the run-time library of the Fortran compiler in use.

By preceding the desired format code by the string ‘MATLAB’, nag_file_print_matrix_real_gen_comp (x04cb) will print the matrix such that it can be input into MATLAB, and **title** will be used as the name of the matrix.

Examples of valid values for **form** are 'F11.4', '1PE13.5', 'G14.5', 'MATLABF11.4', 'MATLAB$*$'.

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

5: **title** – CHARACTER(*)

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.

6: **labrow** – CHARACTER(1)

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'.

7: **rlabs**(:) – CHARACTER(*) array

The dimension of the array **rlabs** must be at least **m** if **labrow** = 'C', and at least 1 otherwise

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**.

8: **labcol** – CHARACTER(1)

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'.

9: **clabs**(:) – CHARACTER(*) array

The dimension of the array **clabs** must be at least **n** if **labcol** = 'C', and at least 1 otherwise

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.

10: **ncols** – INTEGER

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.

11: **indent** – INTEGER

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.

5.2 Optional Input Parameters

1: **m** – INTEGER
2: **n** – INTEGER

Default: the first dimension of the array **a** and the second dimension of the array **a**.

The number of rows and columns of the matrix, respectively, to be printed.

If either **m** or **n** is less than 1, nag_file_print_matrix_real_gen_comp (x04cb) will exit immediately after printing **title**; no row or column labels are printed.

5.3 Output Parameters

1: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

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 $= 4$

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.

ifail $= -399$

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

ifail $= -999$

Dynamic memory allocation failed.

7 Accuracy

Not applicable.

8 Example

This example calls nag_file_print_matrix_real_gen_comp (x04cb) twice, to print matrices of different orders and bandwidths; various options for labelling and formatting are illustrated.

8.1 Program Text

```
     function x04cb_example

fprintf('x04cb example results\n\n');

nmax = 5;
a = zeros(nmax,nmax);

for i = 1:nmax
  a(i,:) = [1:nmax] + 10*i;
end

% Example 1: General 3x5 matrix with non-unit diagonal
mtitle  = 'Example 1:';
matrix = 'General';
diag = ' ';

% Use default formating
format = ' ';

% Integer Labels
labrow = 'Integer';
labcol = 'Integer';
rlabs  = {'       '};
clabs  = {'       '};

% 80 columns wide and no left margin
ncols  = nag_int(80);
indent = nag_int(0);

[ifail] =  x04cb(...
                 matrix, diag, a(1:3,:), format, mtitle, labrow, ...
                 rlabs, labcol, clabs, ncols, indent);

fprintf('\n');
% Example 2: Non-unit upper triangular 5x5 matrix
mtitle  = 'Example 2:';
matrix = 'Upper';
diag = 'Non-unit';

% Use '%8.2f' style formatting formating
format = 'F8.2';

% Character Labels
labrow = 'Character';
labcol = labrow;
rlabs  = {'Uno    '; 'Due    '; 'Tre    '; 'Quattro'; 'Cinque '};
clabs  = {'Un     '; 'Deux   '; 'Trois  '; 'Quatre '; 'Cinq   '};

[ifail] =  x04cb(...
                 matrix, diag, a, format, mtitle, labrow, ...
                 rlabs, labcol, clabs, ncols, indent);

fprintf('\n Example 3:\n');
% Third matrix: Lower triangular 3x3 non-unit matrix, Matlab format
mtitle = 'A';
matrix = 'Lower';
diag   = 'Non-unit';
format = 'MATLABF8.2';
labrow = 'No labels';
labcol = labrow;

[ifail] =  x04cb(...
                 matrix, diag, a(1:3,1:3), format, mtitle, labrow, ...
                 rlabs, labcol, clabs, ncols, indent);
```

8.2 Program Results

```
    x04cb example results

 Example 1:
            1          2          3          4          5
 1     11.0000    12.0000    13.0000    14.0000    15.0000
 2     21.0000    22.0000    23.0000    24.0000    25.0000
 3     31.0000    32.0000    33.0000    34.0000    35.0000

 Example 2:
              Un     Deux   Trois  Quatre    Cinq
     Uno    11.00    12.00   13.00   14.00   15.00
     Due             22.00   23.00   24.00   25.00
     Tre                     33.00   34.00   35.00
 Quattro                             44.00   45.00
  Cinque                                     55.00

 Example 3:
 A = [
     11.00    0.00    0.00;
     21.00   22.00    0.00;
     31.00   32.00   33.00;
     ];
```