NAG Toolbox

nag_file_print_matrix_real_band_comp (x04cf)

1 Purpose

nag_file_print_matrix_real_band_comp (x04cf) prints a double band matrix stored in a packed two-dimensional array.

2 Syntax

```
[ifail] = nag_file_print_matrix_real_band_comp(m, n, kl, ku, a, form, title, labrow, rlabs, labcol, clabs, ncols, indent)
[ifail] = x04cf(m, n, kl, ku, a, form, title, labrow, rlabs, labcol, clabs, ncols, indent)
```

3 Description

nag_file_print_matrix_real_band_comp (x04cf) prints a double band matrix stored in a packed two-dimensional array, 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: **m** – INTEGER

2: **n** – INTEGER

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

If either **m** or **n** is less than 1, nag_file_print_matrix_real_band_comp (x04cf) will exit immediately after printing **title**; no row or column labels are printed.

3: **kl** – INTEGER

The number of subdiagonals of the band matrix A.

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

4: **ku** – INTEGER

The number of superdiagonals of the band matrix A.

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

5: $\mathbf{a}(lda,:)$ – REAL (KIND=nag_wp) array

The first dimension of the array **a** must be at least $\mathbf{kl}+\mathbf{ku}+1$.

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

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

$$\mathbf{a}(k_u + 1 + i - j, j) \quad \text{for } \max(1, j - k_u) \leq i \leq \min(m, j + k_l).$$

6: **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_band_comp (x04cf) to choose its own format code:

form = ' '
nag_file_print_matrix_real_band_comp (x04cf) 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_band_comp (x04cf) 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_band_comp (x04cf) 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.

7: **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.

8: **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'.

9: **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**.

10: **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'.

11: **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.

12: **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.

13: **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

None.

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, **kl** < 0.

ifail $= 2$

On entry, **ku** < 0.

ifail $= 3$

On entry, $lda <$ **kl** $+$ **ku** $+ 1$.

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 Further Comments

None.

9 Example

This example calls nag_file_print_matrix_real_band_comp (x04cf) three times, to print 5 by 5 matrices of different bandwidths; various options for labelling and formatting are illustrated.

9.1 Program Text

```
     function x04cf_example

fprintf('x04cf example results\n\n');

m = nag_int(5);
n = nag_int(5);

% First Example: tridiagonal matrix A
mtitle  = 'Example 1:';
kl = nag_int(1);
ku = nag_int(1);
a = [ 0, 12, 13, 14, 15;
     21, 22, 23, 24, 25;
     31, 32, 33, 34,  0];

% Default format
format = ' ';

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

% 80 columns wide, no indent
ncols  = nag_int(80);
indent = nag_int(0);

[ifail] = x04cf( ...
                 m, n, kl, ku, a, format, mtitle, labrow, ...
                 rlabs, labcol, clabs, ncols, indent);

fprintf('\n');
% First Example: 2 Super-diagonals + 1 sub-diagonal
mtitle  = 'Example 2:';
kl = nag_int(1);
ku = nag_int(2);
a = [ 0,  0, 13, 14, 15;
      0, 22, 23, 24, 25;
     31, 32, 33, 34, 35;
     41, 42, 43, 44,  0];

% '%8.2f' style format
format = 'F8.2';

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

[ifail] = x04cf( ...
                 m, n, kl, ku, a, format, mtitle, labrow, ...
                 rlabs, labcol, clabs, ncols, indent);

fprintf('\n Example 3:\n');
% Third matrix: As above but printed in Matlab form
mtitle = 'A';
format = 'MATLABF8.2';
labrow = 'No labels';
labcol = labrow;

[ifail] = x04cf( ...
                 m, n, kl, ku, a, format, mtitle, labrow, ...
                 rlabs, labcol, clabs, ncols, indent);
```

9.2 Program Results

```
     x04cf example results

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

Example 2:
              Un    Deux   Trois  Quatre    Cinq
    Uno    31.00   22.00   13.00
    Due    41.00   32.00   23.00   14.00
    Tre            42.00   33.00   24.00   15.00
Quattro                    43.00   34.00   25.00
 Cinque                            44.00   35.00

Example 3:
A = [
    31.00   22.00   13.00    0.00    0.00;
    41.00   32.00   23.00   14.00    0.00;
     0.00   42.00   33.00   24.00   15.00;
     0.00    0.00   43.00   34.00   25.00;
     0.00    0.00    0.00   44.00   35.00;
    ];
```