

6.2 Extended Vector and Matrix Output

A. Purpose

The subroutines described here print vectors and matrices in an attractive format. They differ from those in Chapter 6.1 primarily by allowing the user to indicate the length of the line, the unit number, and the number of significant digits to print.

B. Usage

B.1 Program Prototype, Vector Output, Single Precision

INTEGER N , **LWIDTH**, **LUNIT**, **NUMDIG**

REAL V(\geq N)

Assign values to all subroutine arguments

```
CALL SVECPR(V, N, 'Text', LWIDTH,
            LUNIT, NUMDIG)
```

The Vector in V() will now have been printed.

B.1.a Argument Definitions

V() [in] The vector to be printed.

N [in] The number of elements in the vector V().

'Text' [in] Text to be printed as a heading for the output. If the first character is "0" it is used for Fortran vertical format control, *i.e.* it gives an extra blank line, else the first character is treated as part of the text.

LWIDTH [in] The width of the output line. Fortran vertical format control is not counted as part of the line. Note that on many CRT displays that have 80 character lines, if exactly 80 characters are printed an extra blank line is generated. (If $LWIDTH \leq 20$, the current default line width is used; see the message routine MESS, Chapter 19.3.)

LUNIT [in] Fortran unit number for the output. If LUNIT is 0 output is printed on the standard output unit with Fortran vertical format control, else it is printed to a file without Fortran vertical format control.

(If $LUNIT < 0$, the current default is used; see the message routine MESS.)

NUMDIG [in] Number of digits to print for each item. (If $NUMDIG \leq 0$, the current default is used, which is the full floating point precision unless this default has been changed with a call to the message routine MESS.)

B.2 Program Prototype, Matrix Output, Single Precision

INTEGER IDIMA, M, N, **LWIDTH**, **LUNIT**,
NUMDIG

REAL A(IDIMA, \geq N)

Assign values to all subroutine arguments

```
CALL SMATPR(A, IDIMA, M, N, 'Text',
            LWIDTH, LUNIT, NUMDIG)
```

The Matrix in A will now have been printed.

B.2.a Argument Definitions

A(,) [in] The matrix to be printed.

IDIMA [in] The declared first dimension of A(,).

M [in] The number of rows from A(,) to be printed.

N [in] The number of columns from A(,) to be printed.

'Text' [in] Text to be printed as a heading for the output, as for SVECPR above.

LWIDTH, **LUNIT**, **NUMDIG** [in] all are defined as for SVECPR above.

B.3 Modifications for Double Precision

For double precision usage change the names SVECPR and SMATPR to DVECPR and DMATPR, and change the REAL declarations to DOUBLE PRECISION.

B.4 Modifications for Integer

For integer usage change the names SVECPR and SMATPR to IVECPR and IMATPR, change the REAL declarations to INTEGER, and omit the argument NUMDIG.

C. Examples and Remarks

The programs DRVECPR and DRMATPR illustrate the use of these routines to print vectors and matrices respectively. Output is shown in ODVECPR and ODMATPR.

D. Functional Description

The contents of the character string are always printed. If no heading is desired, use ' '. If one wants the heading on a separate line from the first line of the vector output, one should pad the text with enough blanks to fill all but the last character of the line. Numbers are printed using the fewest number of characters possible, subject to keeping the numbers from one line to the next aligned, and in the case of floating point, printing all the digits requested.

The vector output prints “ $i - j$: ” at the beginning of every line after the first, where i is the index for the first component of the vector printed on that line, and j the index of the last. If there is more than one line, the first line is offset so that numbers on following lines line up with those on the first line.

The matrix output prints “Col i ” at the head of each column, and “Row j ” at the left of each row, where i is the index for the column and j is the index for the row.

These routines are short routines that call the message processors described in Chapter 19.3. More flexibility in the printing of text can be obtained by calling SMESS, DMESS, or MESS directly.

E. Error Procedures and Restrictions

No vector is printed if $N \leq 0$, and no matrix is printed if N , M , or $IDIMA \leq 0$.

F. Supporting Information

The source language is ANSI Fortran 77.

Algorithm and code due to F. T. Krogh, JPL, November 1991.

Entry	Required Files
DMATPR	AMACH, DMATPR, DMESS, MESS
DVECPR	AMACH, DMESS, DVECPR, MESS
IMATPR	IMATPR, MESS
IVECPR	IVECPR, MESS
SMATPR	AMACH, MESS, SMATPR, SMESS
SVECPR	AMACH, MESS, SMESS, SVECPR

DRVECPR

```

c      DRVECPR
c>> 2006-04-02 DRVECPR Krogh Added checks with exponents.
c>> 2001-05-22 DRVECPR Krogh Minor change for making .f90 version.
c>> 1991-11-22 DRVECPR Krogh Initial Code.
c
      integer LWIDTH, LUNIT, NUMDIG, N, I
      parameter (LWIDTH=79)
      parameter (LUNIT=0)
      parameter (NUMDIG = 0)
      parameter (N = 6)
      integer A(N)
      real B1(N), B2(N)
      double precision C1(N), C2(N), C3(N)
c
      data (A(I), I=1,N) / 1,2,3,4,5,6 /
      data (B1(I), I=1,N) / 1.E0,2.E0,3.E0,4.E0,5.E0,6.E0/
      data (B2(I), I=1,N) / 1.E4,2.E-4,-3.E0,4.E0,5.E0,6.E0/
      data (C1(I), I=1,N) / 1.D2,2.D0,3.D0,-4.D1,5.D0,6.D0 /
      data (C2(I), I=1,N) / 1.D2,2.D0,3.D0,-4.D11,5.D0,6.D0 /
      data (C3(I), I=1,N) / 1.D14,2.D1,-3.D100,4.D0,5.D0,6.D3/
c
      call IVECPR (A,N,'0 IVECPR.. A()=', LWIDTH, LUNIT)
      call SVECPR (B1,N,'0 SVECPR.. B1()=', LWIDTH, LUNIT, NUMDIG)
      call DVECPR (C1,N,'0 DVECPR.. C1()=', LWIDTH, LUNIT, NUMDIG)
      call SVECPR (B2,N,'0 SVECPR.. B2()=', LWIDTH, LUNIT, NUMDIG)
      call DVECPR (C2,N,'0 DVECPR.. C2()=', LWIDTH, LUNIT, NUMDIG)
      call DVECPR (C3,N,'0 DVECPR.. C3()=', LWIDTH, LUNIT, NUMDIG)
c
      end

```

ODVECPR

```
IVECPR.. A()= 1 2 3 4 5 6
```


DRMATPR

```

c      DRMATPR
c>> 2006-04-02 DRMATPR Krogh Added checks with exponents.
c>> 2001-05-22 DRMATPR Krogh Minor change for making .f90 version.
c>> 1991-11-22 DRMATPR Krogh Initial Code.
c
      integer LWIDTH, LUNIT, NUMDIG, M, N, I, J
      parameter (LWIDTH=79)
      parameter (LUNIT=0)
      parameter (NUMDIG = 0)
      parameter (M = 4)
      parameter (N = 3)
c
      integer A(M,N)
      real B(M,N)
      double precision C1(M,N), C2(M,N), C3(M,N)
c
      data ((A(I, J), J=1,N), I=1,M) / 1,2,3,4,5,6,7,8,9,10,11,12 /
      data ((B(I, J), J=1,N), I=1,M) / 1.,2.,3.,4.,5.,6.,7.,8.,9.,
*                                     10.,11.,12. /
      data ((C1(I, J), J=1,N), I=1,M) / 1.D0,2.D0,3.D0,4.D0,5.D0,
*                                     6.D0,7.D0,8.D0,9.D0,10.D0,
*                                     11.D0,12.D0 /
      data ((C2(I, J), J=1,N), I=1,M) / 1.D0,2.D0,3.D0,4.D0,5.D0,
*                                     6.D0,7.D0,8.D99,9.D0,10.D0,
*                                     -11.D0,12.D0 /
      data ((C3(I, J), J=1,N), I=1,M) / 1.D0,2.D0,3.D0,4.D0,5.D0,
*                                     6.D0,7.D0,8.D123,9.D0,10.D0,
*                                     11.D0,12.D0 /
c
      call IMATPR (A,M,M,N, '0 IMATPR.. A(,)= ', LWIDTH, LUNIT)
      call SMATPR (B,M,M,N, '0 SMATPR.. B(,)= ', LWIDTH, LUNIT, NUMDIG)
      call DMATPR (C1,M,M,N, '0 DMATPR.. C1(,)= ', LWIDTH, LUNIT, NUMDIG)
      call DMATPR (C2,M,M,N, '0 DMATPR.. C2(,)= ', LWIDTH, LUNIT, NUMDIG)
      call DMATPR (C3,M,M,N, '0 DMATPR.. C3(,)= ', LWIDTH, LUNIT, NUMDIG)
      stop
c
      end

```

ODMATPR

```

IMATPR.. A(,)=
      Col 1   Col 2   Col 3
Row 1         1         2         3
Row 2         4         5         6
Row 3         7         8         9
Row 4        10        11        12

SMATPR.. B(,)=
      Col 1           Col 2           Col 3
Row 1  1.0000000    2.0000000    3.0000000
Row 2  4.0000000    5.0000000    6.0000000
Row 3  7.0000000    8.0000000    9.0000000
Row 4 10.0000000   11.0000000   12.0000000

```

```

DMATPR.. C1(,)=
      Col 1          Col 2          Col 3
Row 1  1.0000000000000000  2.0000000000000000  3.0000000000000000
Row 2  4.0000000000000000  5.0000000000000000  6.0000000000000000
Row 3  7.0000000000000000  8.0000000000000000  9.0000000000000000
Row 4 10.0000000000000000 11.0000000000000000 12.0000000000000000

```

```

DMATPR.. C2(,)=
      Col 1          Col 2          Col 3
Row 1  1.0000000000000000E+00  2.0000000000000000E+00  3.0000000000000000E+00
Row 2  4.0000000000000000E+00  5.0000000000000000E+00  6.0000000000000000E+00
Row 3  7.0000000000000000E+00  8.0000000000000000E+99  9.0000000000000000E+00
Row 4  1.0000000000000000E+01 -1.1000000000000000E+01  1.2000000000000000E+01

```

```

DMATPR.. C3(,)=
      Col 1          Col 2          Col 3
Row 1  1.0000000000000000E+000  2.0000000000000000E+000  3.0000000000000000E+000
Row 2  4.0000000000000000E+000  5.0000000000000000E+000  6.0000000000000000E+000
Row 3  7.0000000000000000E+000  8.0000000000000000E+123  9.0000000000000000E+000
Row 4  1.0000000000000000E+001  1.1000000000000000E+001  1.2000000000000000E+001

```