

16.2 Character-based Graphics — Single Print Line

A. Purpose

This subroutine constructs a character string that may be printed as part of a single line image by the user's program to give a graphical representation of data. It is intended primarily for use as a supplement to ordinary tabular output of data as an aid in spotting trends, wild points, etc.

B. Usage

B.1 Program Prototype, Single precision

INTEGER NCHAR

REAL Y, Y1, Y2

LOGICAL RESET

CHARACTER*1 SYMBOL

CHARACTER*n IMAGE [$n \geq$ NCHAR]

Assign values to Y, SYMBOL, NCHAR, Y1, Y2, and RESET.

CALL SPRPL (Y, SYMBOL, IMAGE,
NCHAR, Y1, Y2, RESET)

On return the string, IMAGE, contains the character SYMBOL positioned as a function of the value of Y. The user's program may then print IMAGE as part of a line containing Y and possibly other information.

B.2 Argument Definitions

Y [in] Data value to be plotted. Y should be between Y1 and Y2; otherwise see Section E, Error Procedures.

SYMBOL [in] A single character to be used as a plot symbol. The character can be specified literally in the call statement, as for example: CALL SPRPL (Y, '!', ...)

IMAGE [inout] Character string in which a plot image is constructed.

NCHAR [in] Number of character positions in IMAGE to be used in constructing the plot image. Require $NCHAR \geq 2$.

Y1,Y2 [in] Numbers that bracket the range of values of Y to be plotted in IMAGE. Either $Y1 \leq Y2$ or $Y1 \geq Y2$ is acceptable.

RESET [in] Flag to reset the line image. If RESET = .TRUE., the subroutine will:

1. Store NCHAR blank characters into IMAGE, and then store the character '0' in the zero value position if zero is contained in the interval

2. Store the character specified by SYMBOL in the Y value position.

If RESET = .FALSE., the subroutine will only execute Step 2 above.

B.3 Modifications for Double Precision

For double-precision usage change the REAL type statement to DOUBLE PRECISION, and change the subroutine name from SPRPL to DPRPL.

C. Examples and Remarks

Print a set of (x, y) -data. On the right side of the page print a "strip chart" plot of the data with x increasing downward and y increasing to the right. See DRSPRPL and ODSRPL for code and output illustrating this example.

D. Functional Description

1. The subroutine will compute $ymin = \min(Y1, Y2)$ and $ymax = \max(Y1, Y2)$. Then, if $ymin = ymax$ these numbers will be replaced by $ymin = 0.9 \times ymin$ and $ymax = 1.1 \times ymax$ if $ymin \neq 0.$, or by $ymin = -1$, and $ymax = +1$, if $ymin = 0$.
2. If zero is not in the interval $[ymin, ymax]$, then the scaling will be such that $ymin$ corresponds to the center of the leftmost character position and $ymax$ corresponds to the center of the rightmost character position. If zero is in the interval $[ymin, ymax]$, and does not correspond to the first or last character position in IMAGE(), then scaling will be adjusted so the value zero corresponds to the center of a character position. This adjustment guarantees that values located symmetrically with respect to zero will be plotted in character positions symmetrically located with respect to the zero character position.

E. Error Procedures and Restrictions

A Y value outside the stated data range, $[Y1, Y2]$, (plus a small tolerance) will not be plotted, but the message 'OUT' is placed in either the left or right end of IMAGE() as appropriate. This message will be suppressed if $NCHAR < 6$.

F. Supporting Information

The source language is ANSI Fortran 77.

Based on 1969 code by C.L. Lawson, JPL. Adapted for MATH77 by C.L. Lawson and S. Chiu, JPL, 1983. At MATH77 Release 2.2, Nov. 1988, introduced DPRPL,

and changed the name of the previous PRPL to SPRPL. Programs that were using PRPL should be changed to use the name SPRPL.

Entry	Required Files
DPRPL	DPRPL
SPRPL	SPRPL

DRSPRPL

```

c      program DRSPRPL
c>> 1996-07-12 DRDPRPL1 Krogh Special code for C conversion.
c>> 1994-10-19 DRSPRPL
c>> 1989-04-26 DRSPRPL
c—S replaces "?": DR?PRPL, ?PRPL
c      Demo driver for SPRPL.
c
c
c      character*36 IMAGE
c      real          X, Y
c      integer I
c
c
c      print '(10X, ''X'',6X, ''Y'',21X, ''Y'')'
c      do 20002 I=1,21
c         X = real(I-11)/10.0e0
c         Y = 2.0e0 * X * X - 1.0e0
c         call SPRPL(Y, '*', IMAGE, 36, -1.0e0, 1.0e0, .true.)
c%%      printf( " %10.1f%8.2f %36.36s\n", x, y, image );
c      print '(2x,f10.1,f8.2,1x,a)', X,Y,IMAGE
20002 continue
c      stop
c      end

```

ODSPRPL

X	Y	Y		
-1.0	1.00	0		*
-0.9	0.62	0	*	
-0.8	0.28	0	*	
-0.7	-0.02	*		
-0.6	-0.28	*		
-0.5	-0.50	*		
-0.4	-0.68	*		
-0.3	-0.82	*		
-0.2	-0.92	*		
-0.1	-0.98	*		
0.0	-1.00	*		
0.1	-0.98	*		
0.2	-0.92	*		
0.3	-0.82	*		
0.4	-0.68	*		
0.5	-0.50	*		
0.6	-0.28	*		
0.7	-0.02	*		
0.8	0.28	0	*	
0.9	0.62	0	*	
1.0	1.00	0		*