

## 16.1 Character-based Graphics — One or More XY Graphs

### A. Purpose

These subroutines produce an image in a character array that the user can print to produce a printer plot of one or more  $xy$  data sets with titles and numeric grid-line labels. The size of the image is user-specified. SPRPL1/DPRPL1 will process a single  $xy$  data set, whereas SPRPL2/DPRPL2 will process multiple  $xy$  data sets.

### B. Usage

#### B.1 Usage to plot a single $xy$ data set

##### B.1.a Program Prototype, Single Precision

**INTEGER** NP, NLINES, NCHARS, IERR

**REAL** X( $\geq$ NP), Y( $\geq$ NP)

**CHARACTER\*** $n_1$  **TITLE** [ $1 \leq n_1 \leq$  NCHARS]

**CHARACTER\*** $n_2$  **XNAME** [ $1 \leq n_2 \leq$  NCHARS]

**CHARACTER\*** $n_3$  **YNAME** [ $1 \leq n_3 \leq$  NLINES]

**CHARACTER\*** $n_4$  **IMAGE**( $\geq$ NLINES)  
[ $n_4 \geq$  NCHARS]

Assign values to all arguments except IMAGE() and IERR.

**CALL SPRPL1 (X, Y, NP, TITLE, XNAME, YNAME, NLINES, NCHARS, IMAGE, IERR)**

The printer plot image is returned in IMAGE() and the termination status in IERR.

##### B.1.b Argument Definitions

**X()**, **Y()** [in] Arrays of  $(x, y)$  coordinate pairs defining the curve to be plotted.

**NP** [in] Number of  $(x, y)$  points to be plotted.

**TITLE** [in] Character string to be placed above the plot frame as a title for the graph.

**XNAME** [in] Character string to be placed below the plot frame to identify the abscissa variable.

**YNAME** [in] Character string to be placed in a vertical column to the left of the plot frame to identify the ordinate variable.

**NLINES** [in] Number of lines of a plot image to be built in IMAGE(). The array IMAGE() must be dimensioned at least NLINES.

**NCHARS** [in] Number of character positions to be used in each line in building the plot image in IMAGE(). The declared character length of each element of IMAGE() must be at least NCHARS.

**IMAGE()** [out] Character array in which the printer plot image is built. The image will use NLINES positions in the vertical direction and NCHARS positions in the horizontal direction.

**IERR** [out] Termination status indicator.

0 No errors.

1 NCHARS is too small.

2 NLINES is too small.

#### B.2 Usage to plot multiple $xy$ data sets

##### B.2.a Program Prototype, Single Precision

**INTEGER** IDIM, KC, NR, JX( $\geq$ KC), JY( $\geq$ KC),  
NP( $\geq$ KC)

**INTEGER** NLINES, NCHARS, IERR

**REAL** XY(IDIM,  $m$ ) [ $m \geq \max_{1 \leq k \leq KC} \{JX(k), JY(k)\}$ ]

**CHARACTER\***1 **SYMBOL**( $\geq$ KC)

**CHARACTER\*** $n_1$  **TITLE** [ $1 \leq n_1 \leq$  NCHARS]

**CHARACTER\*** $n_2$  **XNAME** [ $1 \leq n_2 \leq$  NCHARS]

**CHARACTER\*** $n_3$  **YNAME** [ $1 \leq n_3 \leq$  NLINES]

**CHARACTER\*** $n_4$  **IMAGE**( $\geq$ NLINES)  
[ $n_4 \geq$  NCHARS]

Assign values to all arguments except IMAGE() and IERR.

**CALL SPRPL2 (XY, IDIM, KC, JX, JY, NP, SYMBOL, TITLE, XNAME, YNAME, NLINES, NCHARS, IMAGE, IERR)**

The printer plot image is returned in IMAGE() and the termination status in IERR.

##### B.2.b Argument Definitions

**XY(,)** [in] Array of values from which  $(x, y)$  coordinates of points to be placed will be obtained under control of the parameters JX(), JY(), and NP().

**IDIM** [in] Dimension of the first subscript in the XY array. Require  $IDIM \geq \max_{1 \leq k \leq KC} \{NP(k)\}$ .

**KC** [in] Number of  $xy$  sets to be plotted. If  $KC \leq 0$  the subroutine will return taking no action.

**JX()** [in] JX( $k$ ) specifies the column (second subscript) of XY(,) to be used as the  $x$  coordinates for the  $k^{th}$   $xy$  set.

**JY()** [in]  $JY(k)$  specifies the column (second subscript) of  $XY(.)$  to be used as the  $y$  coordinates for the  $k^{th}$   $xy$  set.

**NP()** [in]  $NP(k)$  specifies the number of  $xy$  pairs from  $XY(.)$  in the  $k^{th}$  set to be plotted.

**SYMBOL()** [in]  $SYMBOL(k)$  is the single character to be used for point-plotting the  $k^{th}$  data set.

**TITLE, XNAME, YNAME, NLINES, NCHARS, IMAGE, IERR** Same as in Section B.1 above.

### B.3 Modifications for Double Precision

Change the names `SPRPL1` and `SPRPL2` to `DPRPL1` and `DPRPL2`, respectively, and change the `REAL` declarations to `DOUBLE PRECISION`.

## C. Examples and Remarks

It is permissible, and generally most convenient, to give the arguments `TITLE`, `XNAME`, and `YNAME` as character literals directly in the `CALL` statement. We suggest that no leading or trailing blanks be included in these strings, since each of these strings will be centered in the image based on its length.

The program `DRSPRPL1` and its output `ODSPRPL1` illustrate the use of `SPRPL1` to obtain printer-plots in two different resolutions. The first plot uses `NLINES` = 45 and `NCHARS` = 110, producing a plot that can be displayed on 8.5 inch wide paper using compressed printing, *i.e.*, 16.67 characters per inch. The second plot uses `NLINES` = 22 and `NCHARS` = 79, producing a plot that can be displayed on the  $25 \times 80$  character display frequently used with personal computers. Similarly the use of `SPRPL2` is illustrated by the program `DRSPRPL2` and its output `ODSPRPL2`.

## D. Functional Description

The subroutine first scans the given  $xy$  data to determine maximum and minimum values. It then determines end-values for the  $x$ -axis and a subdivision of the  $x$ -axis into 3 to 10 equal-length subintervals, such that the values at the ends and subdivision points will be representable in decimal with a small number of nonzero digits, and the span of the  $x$ -axis will encompass the given  $x$  data. This same process is applied to the  $y$  data.

The subroutine establishes integer values,  $top$ ,  $bottom$ ,  $left$ , and  $right$ , such that the plot grid will extend vertically from index  $top$  to index  $bottom$  in the `IMAGE()` array, inclusive of the top and bottom grid lines, and horizontally from character position  $left$  to  $right$ , inclusive of the left and right grid lines. These values will satisfy  $1 \leq top < bottom \leq NLINES - 1$ , and  $1 < left < right = NCHARS - 1$ . If `TITLE` is nonblank, it will

be copied into `IMAGE(1)` and  $top$  is set to 2, otherwise  $top = 1$ . If `XNAME` is nonblank, it will be copied into `IMAGE(NLINES)` and  $bottom$  is set to  $NLINES - 2$ , otherwise  $bottom = NLINES - 1$ .  $left$  is set large enough to allow for `YNAME`, if nonblank, to be placed vertically at the left edge of the plot image, for the numeric  $y$ -axis labels to be positioned to the left of the plot grid, and for the leftmost numeric  $x$ -axis label to extend partially to the left of the plot grid. The width of the numeric  $x$ -axis labels, and  $y$ -axis labels are data-dependent.

For minimal useful resolution one should probably arrange to have  $(bottom - top) \geq 20$  and  $(right - left) \geq 20$ , however the subroutine imposes the less stringent requirement that these extents each be at least 10. The subroutine abandons the effort and returns with `IERR`  $\neq 0$  if either of these extents is less than 10.

It is possible that there will not be enough space to place a numeric label below each subdivision point of the  $x$ -axis. The subroutine first places one centered under the left end of the  $x$ -axis. This is assured to be possible by the way  $left$  is defined. It next attempts to place one for the right end of the  $x$ -axis, right justified in the available space. It then sequentially tries to place labels centered under the second, third, etc.,  $x$ -axis subdivision points. It is acceptable if not all  $x$ -axis labels can be placed.

## E. Error Procedures and Restrictions

If the character variable `TITLE` or `XNAME` has length greater than `NCHARS`, only the first `NCHARS` characters are placed in the image. If `YNAME` has length greater than `NLINES`, only the first `NLINES` characters are placed in the image.

This subroutine requires at least enough space for the plot grid boundary indices, defined above in Section D, to satisfy  $(bottom - top) \geq 10$  and  $(right - left) \geq 10$ . If this is not satisfied the subroutine abandons the image-building effort, issues an error message using the error message package of Chapter 19.2 at error level 0, and returns with `IERR` = 1 or 2.

## F. Supporting Information

The source language is ANSI Fortran 77. At Release 4.0 of `MATH77` the subroutines of this Chapter were introduced and previous similar but less general subroutines `PRPL1` and `PRPL2` were deleted.

Based on 1967 code by C. L. Lawson, and J. Hatfield, JPL.

Current version by C. L. Lawson, 1992.

Entry	Required Files	Entry	Required Files
<b>DPRPL1</b>	DPRPL1, DPRPL3, ERFIN, ERMSG, IERM1, IERV1	<b>SPRPL1</b>	ERFIN, ERMSG, IERM1, IERV1, SPRPL1, SPRPL3
<b>DPRPL2</b>	DPRPL2, DPRPL3, ERFIN, ERMSG, IERM1, IERV1	<b>SPRPL2</b>	ERFIN, ERMSG, IERM1, IERV1, SPRPL2, SPRPL3

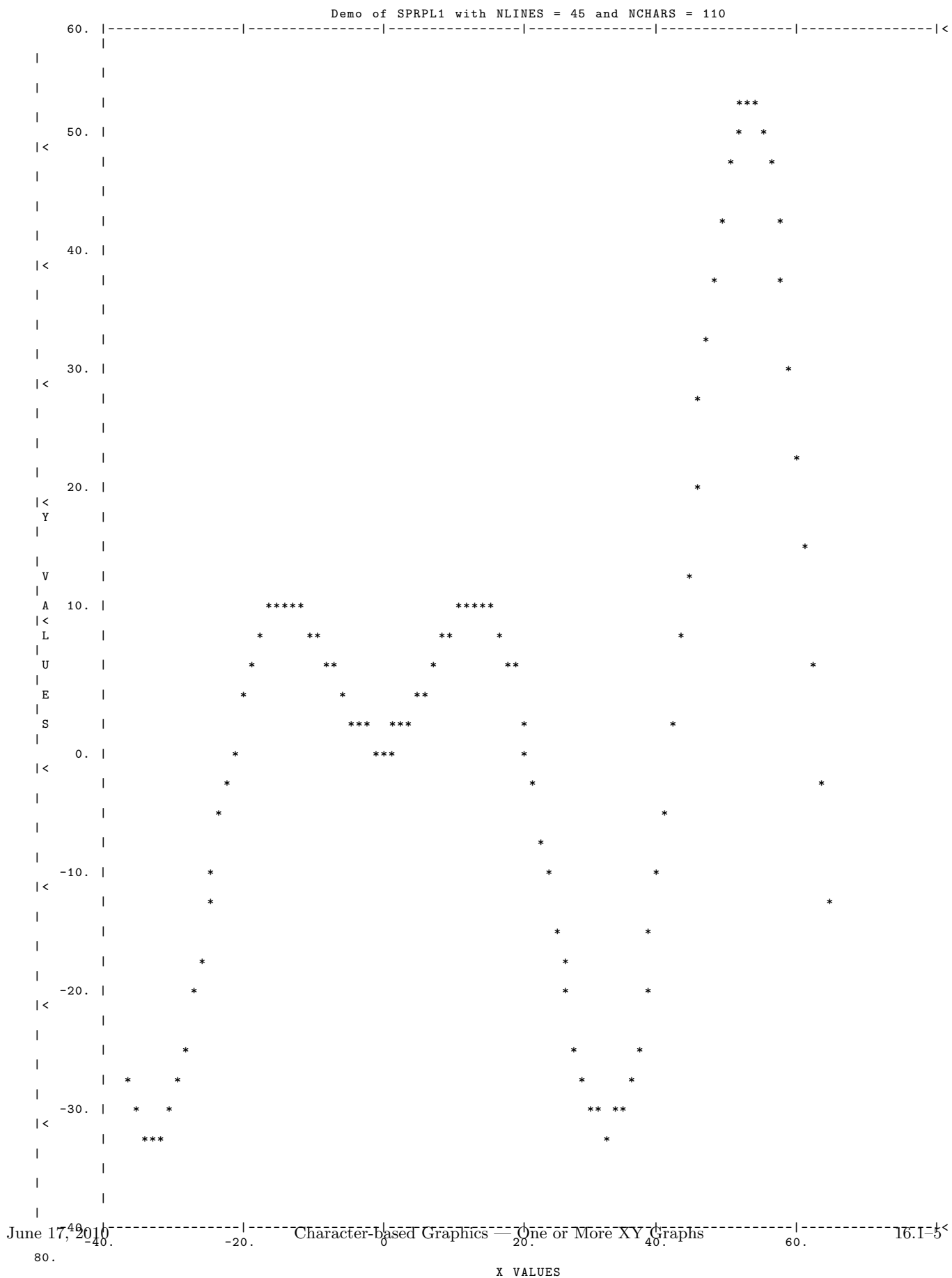
## DRSPRPL1

```

catc      program DRSPRPL1
c>> 1996-06-27 DRSPRPL1 Krogh Special code for C conversion.
c>> 1994-10-19 DRSPRPL1 Krogh Changes to use M77CON
c>> 1992-02-14 DRSPRPL1 CLL
c
c—S replaces "?: DR?PRPL1, ?PRPL1
c
      integer I, IERR, LINE, NLMAX, NCMAX, NP
      parameter(NLMAX = 45, NCMAX = 110, NP = 101)
      real      ALPHA, BETA, DX, X(NP), XT, Y(NP)
      parameter(ALPHA = 0.15e0, BETA = 0.015e0, DX = 1.0e0)
c%%      long int ierr, line;
c%%      char image[45][111], image2[45][80];
      character IMAGE(NLMAX)*(NCMAX)
c
      XT= -36.0e0
      do 10 I=1,NP
          X(I)=XT
          Y(I)=X(I)*SIN(ALPHA*X(I)+BETA)
          XT=XT+DX
      10 continue
c
c++ Code for ~.C. is active
      call SPRPL1(X,Y,NP,
*      'Demo of SPRPL1 with NLINES = 45 and NCHARS = 110',
*      'X VALUES', 'Y VALUES', 45, 110, IMAGE, IERR)
      print '(1x,a)', IMAGE
      print '(1x/1x)'
c
      call SPRPL1(X,Y,NP,
*      'Demo of SPRPL1 with NLINES = 22 and NCHARS = 79',
*      'X VALUES', 'Y VALUES', 22, 79, IMAGE, IERR)
      print '(1x,a)', (IMAGE(LINE))(1:79), LINE = 1,22)
c++ Code for .C. is inactive
c%%      sprpl1( x,y,NP,"Demo of SPRPL1 with NLINES = 45 and NCHARS = 110",
c%%      "X VALUES", "Y VALUES", 45, 111, (byte*)image, &ierr );
c%%      for (line=0; line < 45; line++)
c%%          printf( "%111.111s", image[line] );
c%%      printf( "\n" );
c%%      printf( " \n \n" );
c%%      sprpl1( x,y,NP,"Demo of SPRPL1 with NLINES = 22 and NCHARS = 79",
c%%      "X VALUES", "Y VALUES", 22, 80, (byte*)image2, &ierr );
c%%      printf( " " );
c%%      for (line = 0; line < 22; line++)
c%%          printf( "%80.80s", image2[line] );
c%%      printf( "\n" );
c++ End
      stop
      end

```

## ODSPRPL1



Demo of SPRPL1 with NLINES = 22 and NCHARS = 79

## DRSPRPL2

```

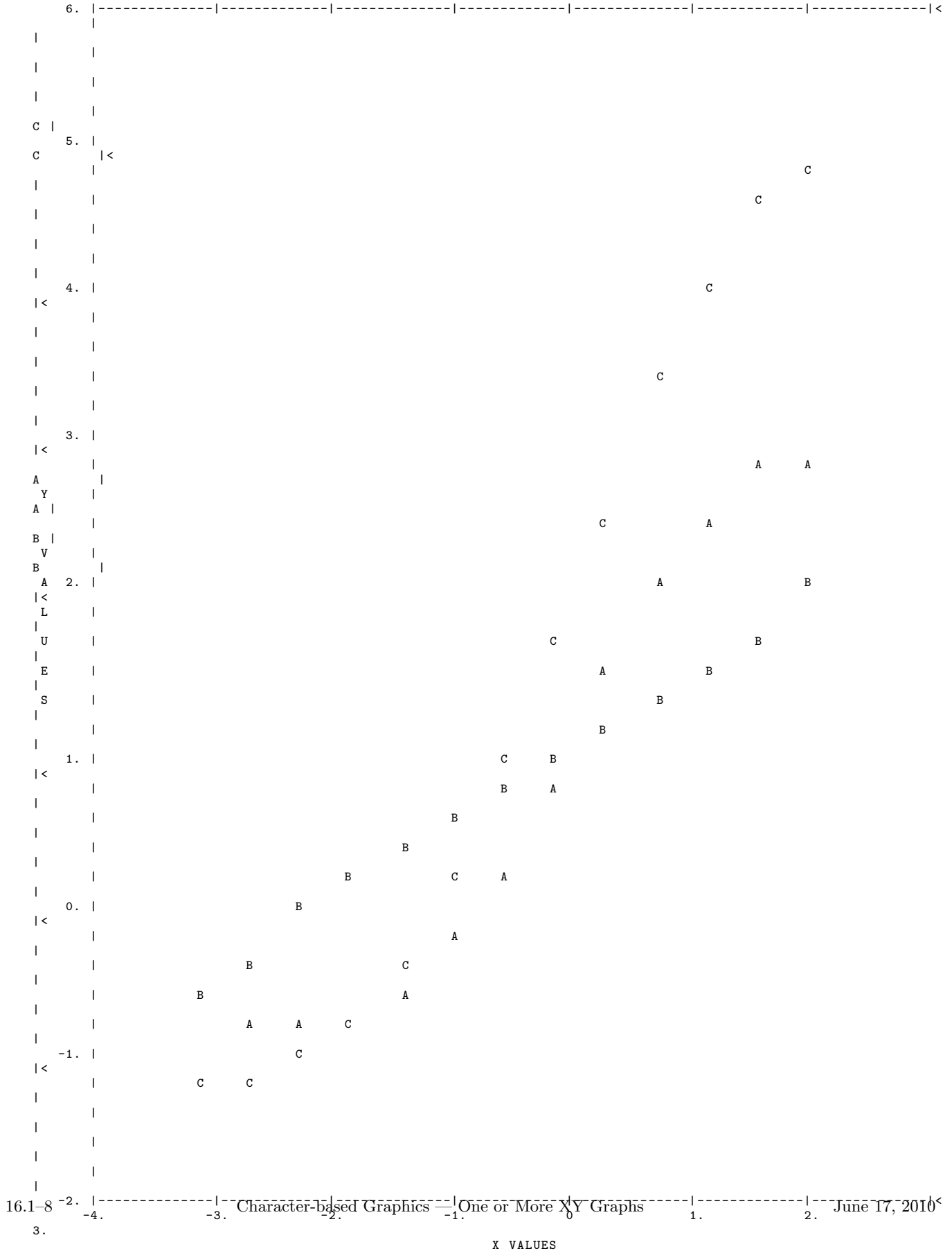
c      program DRSPRPL2
c>> 1996-06-27 DRSPRPL2 Krogh Special code for C conversion.
c>> 1994-10-19 DRSPRPL2 Krogh Changes to use M7CON
c>> 1992-02-14 DRSPRPL2 CLL
c
c—S replaces "?: DR?PRPL2, ?PRPL2
c
      integer I, IERR, KC, LINE, NCMAX, NLMAX
      parameter(KC = 3, NLMAX = 45, NCMAX = 110)
      integer JX(KC), JY(KC), NP(KC)
      real DX, S, X, XY(15,4)
      character SYMBOL(KC)
c%%      long int ierr, line;
c%%      char image[45][111], image2[45][80];
      character IMAGE(NLMAX)*(NCMAX)
      data JX/1,1,1/
      data JY/2,3,4/
      data NP/15,15,15/
      data SYMBOL/'A','B','C'/
c
      X = -3.14e0
      DX = 21.0e0/49.0e0
      do 10 I=1,15
        S = sin(X)
        XY(I,1) = X
        XY(I,2) = S+1.0e0 + 0.5e0*X
        XY(I,3) = 1.0e0 + 0.5e0*X
        XY(I,4) = S + 2.0e0 +X
        X = X+DX
      10 continue

c++ Code for .C. is active
      call SPRPL2(XY,15,KC,JX,JY,NP,SYMBOL,
      * 'Demo of SPRPL2 with NLINES = 45 and NCHARS = 110',
      * 'X VALUES', 'Y VALUES', 45, 110, IMAGE, IERR)
      print '(1x,a)', IMAGE
      print '(1x/1x)',
c
      call SPRPL2(XY,15,KC,JX,JY,NP,SYMBOL,
      * 'Demo of SPRPL2 with NLINES = 22 and NCHARS = 79',
      * 'X VALUES', 'Y VALUES', 22, 79, IMAGE, IERR)
      print '(1x,a)', (IMAGE(LINE)(1:79), LINE = 1,22)
c++ Code for .C. is inactive
c%%      sprpl2( (float *)xy, 15, KC, jx, jy, np, symbol,
c%%      "Demo of SPRPL2 with NLINES = 45 and NCHARS = 110",
c%%      "X VALUES", "Y VALUES", 45, 111, (byte*)image, &ierr );
c%%      for(line=0; line < 45; line++)
c%%        printf( "%111.111s", image[line] );
c%%      printf( " \n\n" );
c%%      sprpl2( (float *)xy, 15, KC, jx, jy, np, symbol,
c%%      "Demo of SPRPL2 with NLINES = 22 and NCHARS = 79",
c%%      "X VALUES", "Y VALUES", 22, 80, (byte*)image2, &ierr );
c%%      for (line = 0; line < 22; line++)
c%%        printf( "%80.80s", image2[line]);
c%%      printf( " \n" );
c++ End
      stop
      end

```

## ODSPRPL2

Demo of SPRPL2 with NLINES = 45 and NCHARS = 110



Demo of SPRPL2 with NLINES = 22 and NCHARS = 79