POSTSCRIPT FOR TECHNICAL DRAWINGS

PSPLOT: A FORTRAN-CALLABLE
POSTSCRIPT PLOTTING LIBRARY

USER'S MANUAL

Kevin E. Kohler

Nova Southeastern University
Oceanographic Center
8000 North Ocean Drive
Dania, Florida 33004

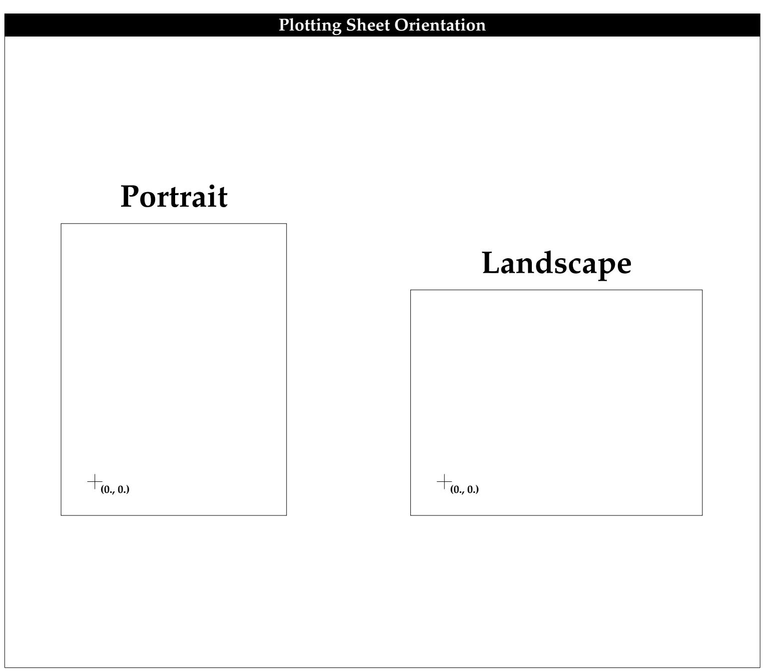
2. PLOTTING BASICS

This section will present the conventions used by the plotting subroutines in the PSPLOT plotting library . This information can be used to create customized plotting subroutines.

A *plotting session* is a set of plotting instructions in a user's application program which produces hardcopy graphic display output. The output can be either a single plot or graph or a set of graphs. **Every plotting session must begin with a call to subroutine PSINIT (with the exception of a call to NEWDEV) and end with a call to subroutine PLOTND.**

Pen movement is occasionally mentioned throughout this manual. While PostScript does not use an actual pen for graphic production, it is useful to visualize the plot commands as directing the movements of a pen of variable thickness to a specified (x,y) coordinate, with the pen being either up or down. The plotting subroutines are simple, user-callable commands which direct the movements of an imagined pen upon a plotting sheet.

The initial coordinate origin is approximately .5 inches from the bottom and left paper edges. The orientation of the page must be specified to be either **portrait** (short side horizontal) or **landscape** (long side horizontal) and is set by the call to PSINIT (which see). The figure below shows the paper orientation and beginning plot origin for portrait and landscape modes. The current plot origin can be relocated to other positions during the plotting session to provide new reference points for subsequent plotting commands. When the next graph is started, the new origin should be placed far enough away to avoid overlapping the just-completed graph.



3. USER-LEVEL PLOTTING ROUTINES

This section describes the subroutines in the PSPLOT plotting library at the time of this writing. These routines are Fortran-callable from an application program. Below is a brief summary of the subroutines, followed by an alphabetical listing of each subroutine containing a more detailed description of its function, syntax, and calling arguments.

ARC	Draws an arc of a specified radius and center.
AROHED	Draws an arrowhead at a specified location.
ARROW	Draws an arrow including the shaft.
AXIS	Draws an axis of a graph, with user-specified axis titles.
BLKSTP	Strips out embedded blanks in a character string.
BORDER	Draws a rectangular border with tick marks.
CHOPIT	Logically closes the current page and begins a new one.
CIRCLE	Draws a circle with specified center, radius and fill.
CLIP	Inserts the PostScript command *clip* into the output file.
CLIPBOX	Defines the current clipping area.
COLBOX	Fills regions with specified color.
CONCOLR	Draws colored contour plots of a 2-dimensional array of regularly-spaced data.
CONFILL	Draws grayscale contour plots of a 2-dimensional array of regularly-spaced data.
CONREC	Draws contour plots of a 2-dimensional array of regularly-spaced data.
CURVE	Draws a curve between 2 points with specified beginning and ending slopes.
DRWCRV	Connects data points with a solid line.
DRWTRI	Draws a triangle.

ONEHLF Draws the symbol ½.

OVERBAR Draws a character string with an overbar.

OVRSBSP Draws an overbar over a subscripted and superscripted character string.

OVRSBSPG Draws an overbar over a subscripted and superscripted Greek symbol.

OVERSUB Draws an overbar over a subscripted character string.

OVERSUBG Draws an overbar over a subscripted Greek symbol.

OVERSUP Draws an overbar over a superscripted character string.

OVERSUPG Draws an overbar over a superscripted Greek symbol.

OVRGRK Draws an overbar over a Greek symbol.

PLOT Directs "pen" movement to a specified (x,y) point with the "pen" either up or down.

PLOTND Closes the output PostScript file and terminates the plotting session.

PLSMIN Draws the ± character.

PRIME Draws the ' character.

PSINIT Called to begin a plotting session. It also specifies portrait or landscape mode.

RECT Draws a rectangle.

RECTFILC Draws a rectangle and fills it with specified red, green, and blue color levels.

RECTFILG Draws a rectangle and fills it with specified gray level.

ROTATE Rotates the current coordinate system by a specified angle. Essentially, ROTATE issues the PostScript *rotate* command.

RRECT Draws a rectangle with rounded corners. It can then fill the rectangle with the current graylevel or RGB values.

SETCOLR Sets the current color values.

ARC

PURPOSE **ARC** draws an arc having a specified radius and arc center.

SYNTAX CALL ARC (XC, YC, RAD, ANG1, ANG2)

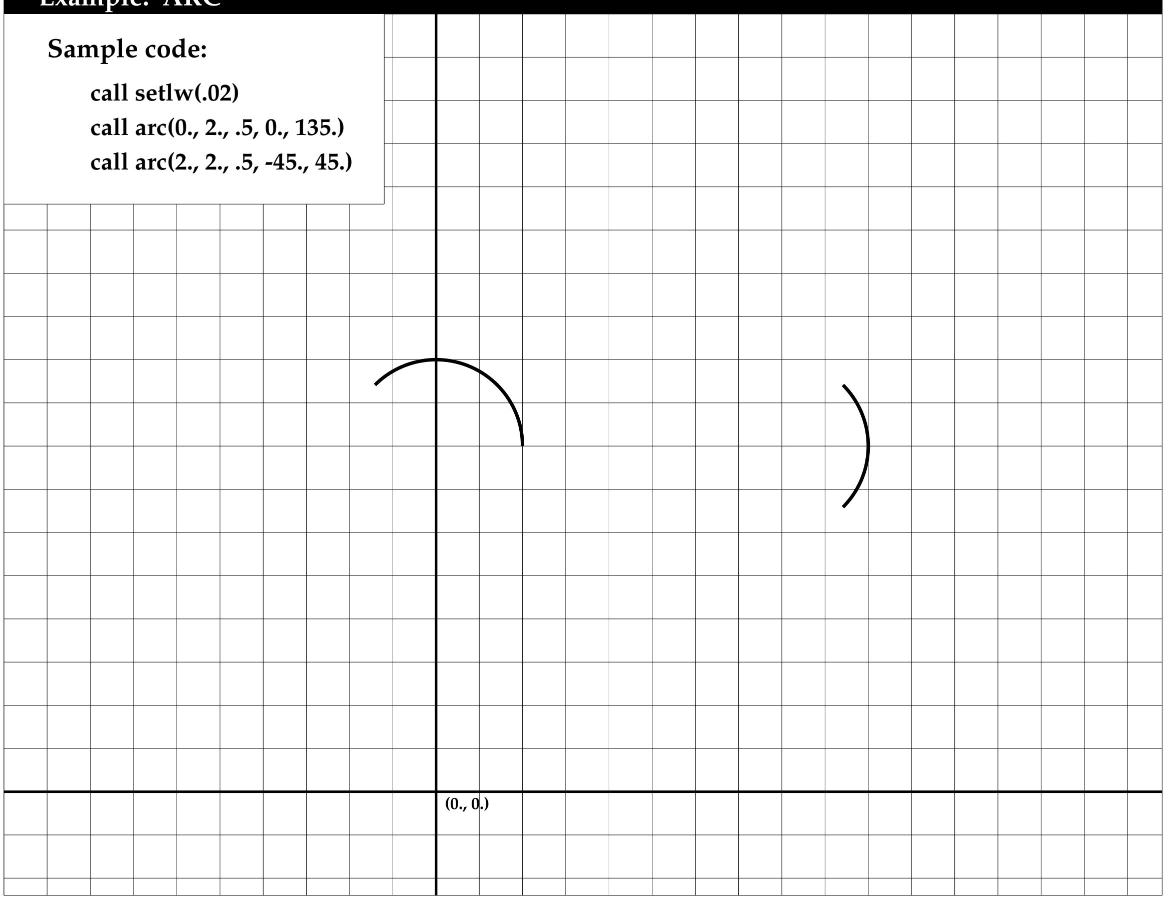
ARGUMENTS

XC, YC	X,Y coordinates of the center of the arc.
RAD	Radius of arc.
ANG1	Angle of arc beginning.
ANG2	Angle of arc ending.

Example: ARC

Sample code:

```
call setlw(.02)
call arc(0., 2., .5, 0., 135.)
call arc(2., 2., .5, -45., 45.)
```



ARROW

PURPOSE **ARROW** draws an arrow, with vector length, at a specified location.

SYNTAX CALL ARROW (XSS, YSS, XPP, YPP, AROLNP, SPRANG, LOCXY)

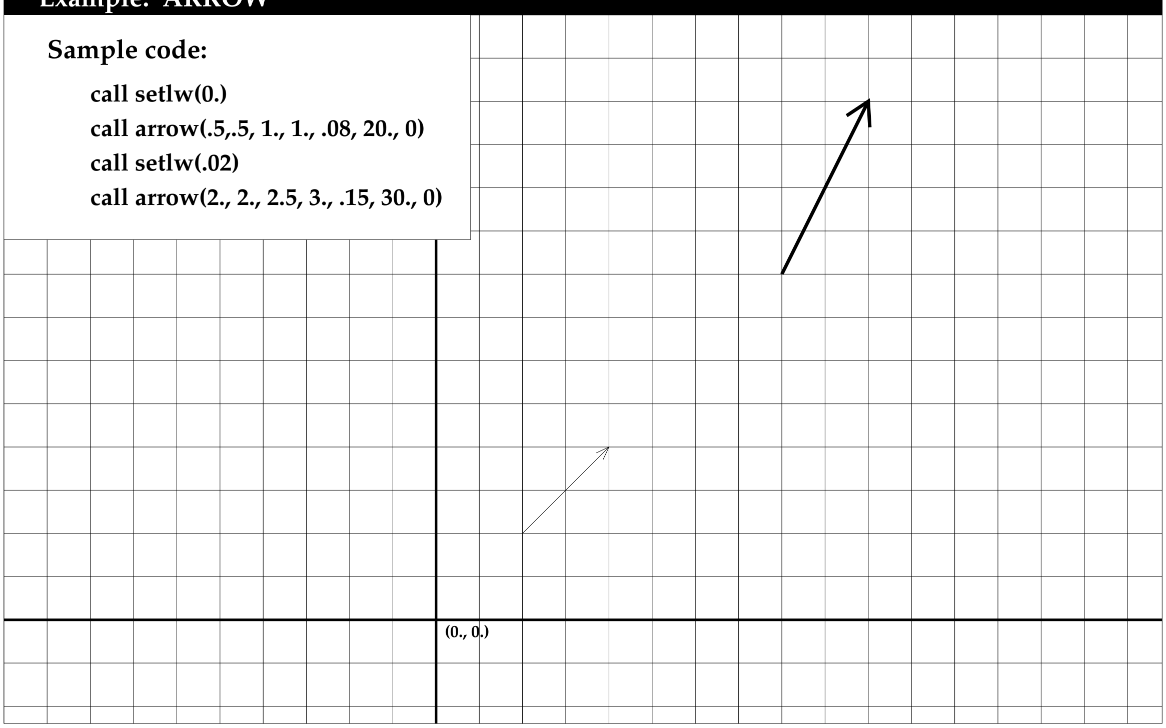
ARGUMENTS

XSS,YSS	X,Y coordinates of the vector length origin.
XPP,YPP	X,Y coordinates of the tip of the arrowhead.
AROLNP	Length of arrowhead sides.
SPRANG	Half the angular spread of arrowhead.
LOCXY	Location of arrowhead point LOCXY=1 XPP,YPP at arrowhead point (most commonly used) LOCXY=2 XPP, YPP at center of arrowhead. LOCXY=3 XPP,YPP at tail of arrowhead.

Example: ARROW

Sample code:

```
call setlw(0.)
call arrow(.5,.5, 1., 1., .08, 20., 0)
call setlw(.02)
call arrow(2., 2., 2.5, 3., .15, 30., 0)
```



BLKSTP

PURPOSE **BLKSTP** strips out embedded blanks in a character string. It returns the new character string and the number of characters in the new string.

SYNTAX CALL BLKSTP (STR1, NDIM, STR2, NC)

ARGUMENTS

STR1	Input character string.
NDIM	Maximum length of STR1 as stated in calling program.
STR2	Output character string with embedded blanks removed.
NC	Number of characters in STR2.

CHOPIT

PURPOSE **CHOPIT** logically closes the current graphics page and begins a new one.

SYNTAX CALL CHOPIT (XPP, YPP)

ARGUMENTS

XPP,YPP X,Y coordinates of the initial plot origin of the next plot. This allows you to begin all plots at the same origin on each page.

Note: XPP and YPP are coordinates independent of the current scaling factor. That is, XPP and YPP represent actual inches, not scaled coordinates. The current scaling factor is reinstated after the new coordinate origin is set.

Subroutine CHOPIT should not be confused with subroutine PLOTND, which is called once at the end of each plotting session and which closes the entire output PostScript file.

CLIP

PURPOSE **CLIP** inserts the PostScript command *clip* into the output file.

SYNTAX CALL CLIP

ARGUMENTS

NONE.

Note: Most uses of this command are bracketed by calls to GSAV and GREST.

COLBOX

PURPOSE **COLBOX** fills a region with a specified color.

SYNTAX CALL COLBOX (XPTS, YPTS, NPTS, RED, GREEN, BLUE)

ARGUMENTS

XPTS	Array containing x-coordinates of region path.
YPTS	Array containing y-coordinates of region path.
NPTS	Number of points in region path.
RED	Red value of specified color in RGB color model.
GREEN	Green value of specified color in RGB color model.
BLUE	Blue value of specified color in RGB color model.

Arguments RED, GREEN, and BLUE must lie between 0. and 1.0, inclusive.

CONFILL

PURPOSE **CONFILL** draws grayscale contour plots of a two-dimensional array of regularly spaced data.

SYNTAX CALL CONFILL(ARR, IMAX, IEXT, JEXT, XLEN, YLEN, CVAL, GRYLEV, NVAL, IOFFP, SPVAL)

<u>*ARGUMENTS*</u>

ARR — Two-dimensional array containing regularly spaced data to be contoured. CONFILL assumes ARR(1,1) is located at the lower left corner of the plot. Data points ARR(1,1) through ARR(IEXT,JEXT) are contoured in an area XLEN x YLEN.

IMAX — The first dimension of ARR in the calling program.

IEXT — Number of points in x-direction of ARR to be contoured.

JEXT — Number of points in y-direction of ARR to be contoured.

XLEN — X-direction length of plotting area.

YLEN — Y-direction length of plotting area.

CVAL — Array containing the values to be used for contour intervals (1 to NVAL). CVAL must be dimensioned with a dimension of at least 1. Unlike subroutine CONREC, NVAL cannot equal 0.

GRYLEV — Array dimensioned (NVAL) containing the grayscale values for each of the contour levels. Regions less than or equal to the corresponding contour value CVAL are filled with the corresponding grayscale values. Grayscale values range from 0. (black) to 1.0 (white).

NVAL — Number of contour intervals. NVAL must be less than or equal to 100, and unlike in CONREC, NVAL cannot equal 0.

IOFFP — Flag indicating that grid boxes whose vertices have the value SPVAL are to be ignored during contouring.

SPVAL — Special value denoting which grid boxes are to be ignored during contouring.

CONREC

PURPOSE **CONREC** draws contour plots of a two-dimensional array of regularly spaced data.

SYNTAX CALL CONREC (ARR, IMAX, IEXT, JEXT, XLEN, YLEN, CVAL, NVAL)

ARGUMENTS

ARR	Two-dimensional array containing regularly spaced data to be contoured. CONREC assumes ARR(1,1) is located at the lower left corner of the plot. Data points ARR(1,1) through ARR(IEXT,JEXT) are contoured in an area XLEN x YLEN.
IMAX	The first dimension of ARR in the calling program.
IEXT	Number of points in x-direction of ARR to be contoured.
JEXT	Number of points in y-direction of ARR to be contoured.
XLEN	X-direction length of plotting area.
YLEN	Y-direction length of plotting area.
CVAL	Array containing the values to be used for contour intervals (1 to NVAL). CVAL must be dimensioned with a dimension of at least 1. If NVAL = 0, the individual contour values are automatically computed.
NVAL	Number of contour intervals. If NVAL = 0, the number of contour intervals and the contour interval values are automatically calculated. This is helpful if the range of data values is not known beforehand. NVAL must be less than or equal to 100.

***Special features of CONREC*:**

If NVAL < 0, only high and lows are plotted instead of contours.

Common block CONPAR is used to control various characteristics of the contour plot from the calling program. The variables in CONPAR are shown below, along with their default values:

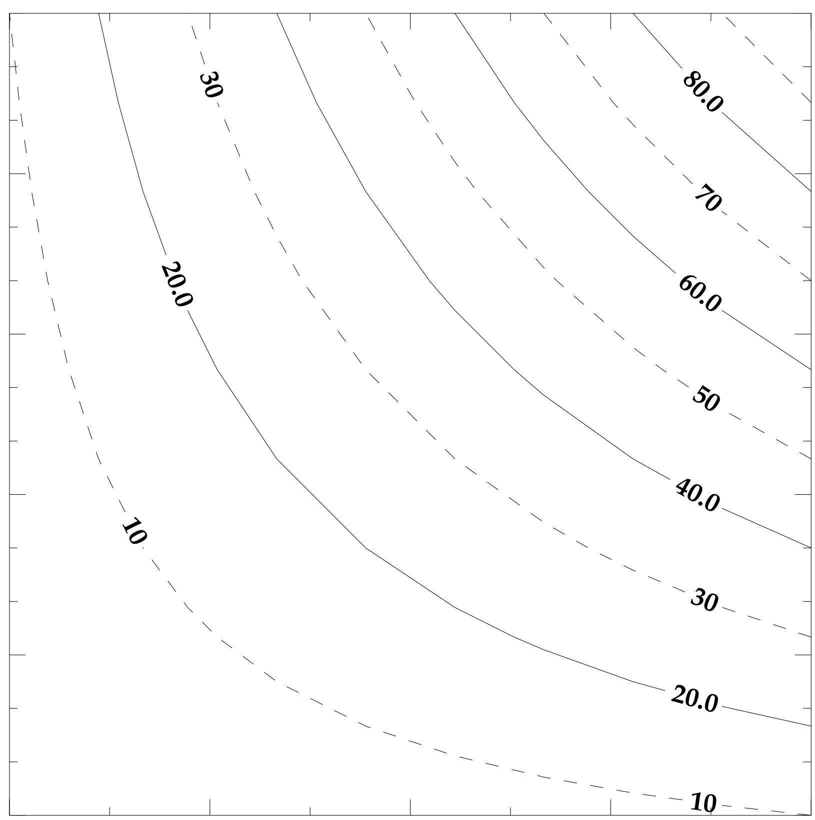
COMMON/CONPAR/ ISPEC, IOFFP, SPVAL, ILEGG, ILABB, NHII, NDECCN, NLBLL, LSCAL, LDASH, HGTLAB

ISPEC If 0, variables in CONPAR have no effect. Default = 0.

Example: CONREC

Sample code:

```
      do 10 j=1,10
      do 10 i=1,10
 10   f(i,j)=i*j
      xlen=5.
      ylen=5.
      nval=6
      call border(xlen, ylen, 1111, 1111, 4, 2, 5, 3)
      do 20 n=1,nval
      cval(n)=(n-1)*20
 20   continue
      nlbll=0
      lscal=1
      call conrec(f, 10, 10, 10, xlen, ylen, cval, nval)
      ldash=6
      ndecn=-1
      do 25 n=1,nval
      cval(n)=10.+(n-1)*20
 25   continue
      call conrec(f, 10, 10, 10, xlen, ylen, cval, nval)
```



DRWCRV

PURPOSE **DRWCRV** draws a solid curve.

SYNTAX CALL DRWCRV (XARR, YARR, NPTS, THK, CLOSER)

ARGUMENTS

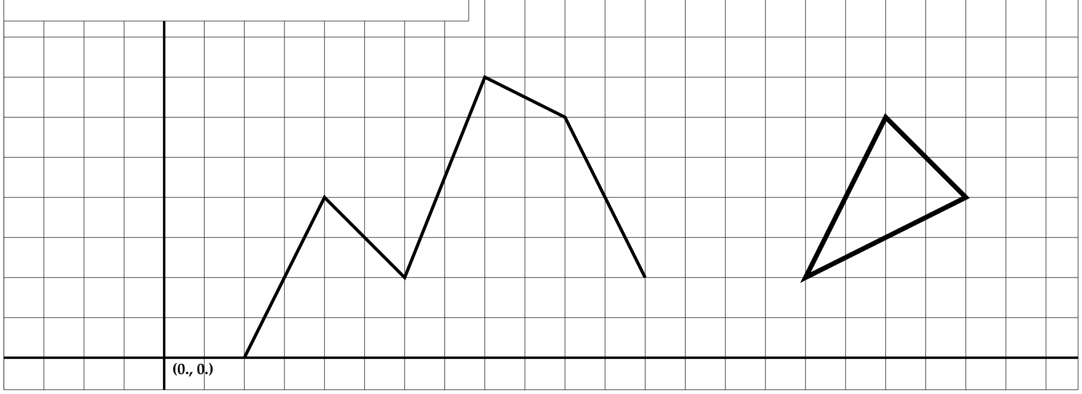
XARR	Array containing the x-coordinates of the points to be connected.
YARR	Array containing the y-coordinates of the points to be connected.
NPTS	Number of points in the curve.
THK	Thickness of the curve. If THK = 0., the current linewidth is used.
CLOSER	Logical variable. If CLOSER = .TRUE., the last point in the data arrays is connected to the first point, i.e. the curve is closed; otherwise, the curve is open.

Example: DRWCRV

Sample code:

```
dimension xp1(6), yp1(6)
dimension xp2(3), yp2(3)
data xp1/ .5, 1., 1.5, 2., 2.5, 3. /
data yp1/ 0., 1., .5, 1.75, 1.5, .5 /
data xp2/ 4., 4.5, 5. /
data yp2/ .5, 1.5, 1. /
call drwcrv(xp1, yp1, 6, .02, .false. )
call drwcrv(xp2, yp2, 3, .03, .true. )
```

Code to draw axes not shown.



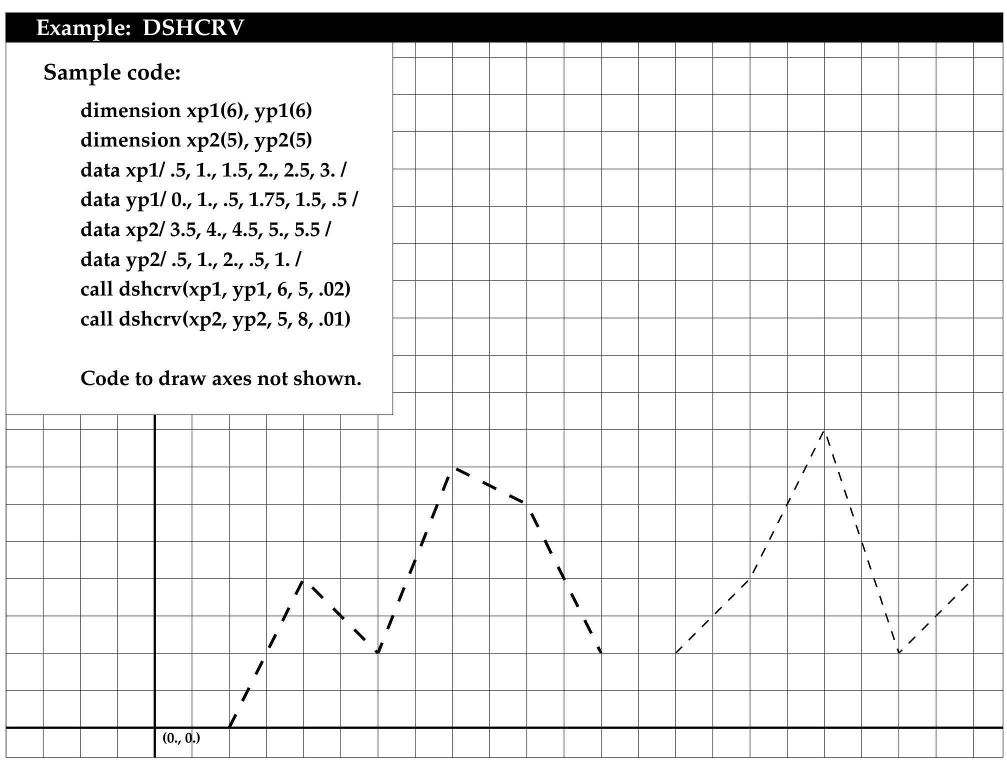
DSHCRV

PURPOSE **DSHCRV** connects points in two data arrays using a line with a specified dash pattern.

SYNTAX CALL DSHCRV (XARR, YARR, NPTS, IDSHPN, THK)

<u>*ARGUMENTS*</u>

XARR	Array containing the x-coordinates of the points to be connected.
YARR	Array containing the y-coordinates of the points to be connected.
NPTS	Number of points in the curve.
IDSHPN	Dashed pattern of the line used to connect the data points. IDSHPN is specified as the number of times the dashed pattern (solid line/blank space) is repeated per inch.
THK	Thickness of the dashes in the curve. If THK = 0., the current line width is used.



FACTOR

PURPOSE **FACTOR** enlarges or reduces the size of a plot by a specified ratio.

SYNTAX CALL FACTOR (FACT)

ARGUMENTS

FACT The ratio of the desired plot size to the actual size. FACT must be a positive floating point number.

For example, to enlarge a plot to twice its normal size:

CALL FACTOR (2.0)

FILLBOX

PURPOSE **FILLBOX** fills a shape with a specified gray level.

SYNTAX CALL FILLBOX (XARR, YARR, NPTS, GRY)

<u>*ARGUMENTS*</u>

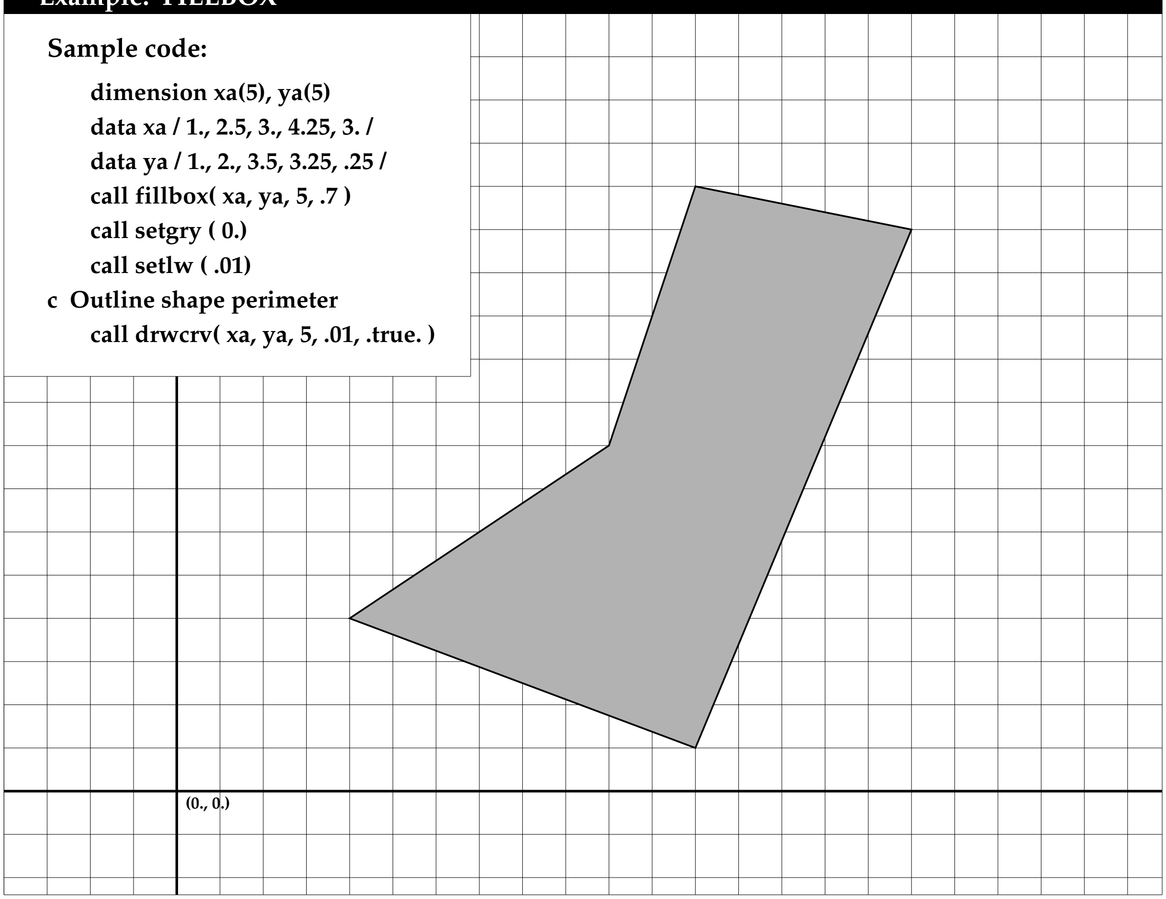
XARR	Array containing the x-coordinates of the points forming the perimeter of the shape to fill.
YARR	Array containing the y-coordinates of the points forming the perimeter of the shape to fill.
NPTS	Number of points comprising the shape perimeter.
GRY	Gray level of filled shape. GRY must lie between 0. (black) and 1. (white), inclusive.

Note: The current gray level remains at value GRY after exiting FILLBOX.

Example: FILLBOX

Sample code:

```
      dimension xa(5), ya(5)
      data xa / 1., 2.5, 3., 4.25, 3. /
      data ya / 1., 2., 3.5, 3.25, .25 /
      call fillbox( xa, ya, 5, .7 )
      call setgry ( 0.)
      call setlw ( .01)
c  Outline shape perimeter
      call drwcrv( xa, ya, 5, .01, .true. )
```



FILRGNC

PURPOSE **FILRGNC** fills a region with the current gray level or color.

SYNTAX CALL FILRGNC (XARR, YARR, NPTS)

ARGUMENTS

XARR	Array containing the X coordinates of the points forming the perimeter of the shape to fill.
YARR	Array containing the Y coordinates of the points forming the perimeter of the shape to fill.
NPTS	Number of points comprising the shape perimeter.

GRKSYM

PURPOSE **GRKSYM** plots a Greek symbol.

SYNTAX CALL GRKSYM (XPP, YPP, HEIGHT, ICH, ANG, NCHAR, MJUS)

ARGUMENTS

XPP,YPP X,Y coordinates of the Greek symbol to be plotted. Plotting may be continued from the end of a previously plotted character when used in conjunction with any of the subroutines which support continuation.

HEIGHT Height of Greek symbol to be plotted.

ICH Integer value between 1 and 50 corresponding to the desired Greek symbol (see table below).

ANG Angle, measured counterclockwise from the X-axis, at which the character is to be plotted.

NCHAR Number of characters to be plotted (should be set to 1).

MJUS Controls the justification of the character to be plotted.

If MJUS = 0, (XPP,YPP) denotes the position of the lower left corner of the plotted character.

If MJUS = 1, (XPP,YPP) denotes the position of the center of the plotted character.

If MJUS = 2, (XPP,YPP) denotes the position of the lower right corner of the plotted character.

PostScript Greek Symbols

Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	ϒ
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Φ	Χ	Ψ	Ω	α	β	γ	δ	ε	ζ	η	ϑ	ι	κ	λ	μ	ν	ξ	ο	π
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
ρ	σ	τ	υ	φ	χ	ψ	ω	θ	ϕ										
41	42	43	44	45	46	47	48	49	50										

HILITEC

PURPOSE **HILITEC** prints text surrounded by a rectangular box (color).

SYNTAX CALL HILITEC (XPP, YPP, HEIGHT, CSTR, ANG, EDG, JUSX, JUSY,
FRED, FGREEN, FBLUE, BRED, BGREEN, BBLUE)

ARGUMENTS

XPP,YPP X,Y coordinates of the box surrounding the text.

HEIGHT Height of text to be plotted.

CSTR Character string text to be plotted.

ANG Angle, measured counterclockwise from the X-axis, at which the text is to be plotted.

EDG Fraction of the text height to use as an edge border.

JUSX Justification in the x-direction.
JUSX=0 is left justification.
JUSX=1 is centered.
JUSX=2 is right justification.

JUSY Similar to JUSX, but in the y-direction.
JUSY=0 is bottom justification.
JUSY=1 is centered.
JUSY=2 is top justification.

FRED, FGREEN, FBLUE Foreground text RGB values. Must lie between 0. and 1.0, inclusive.

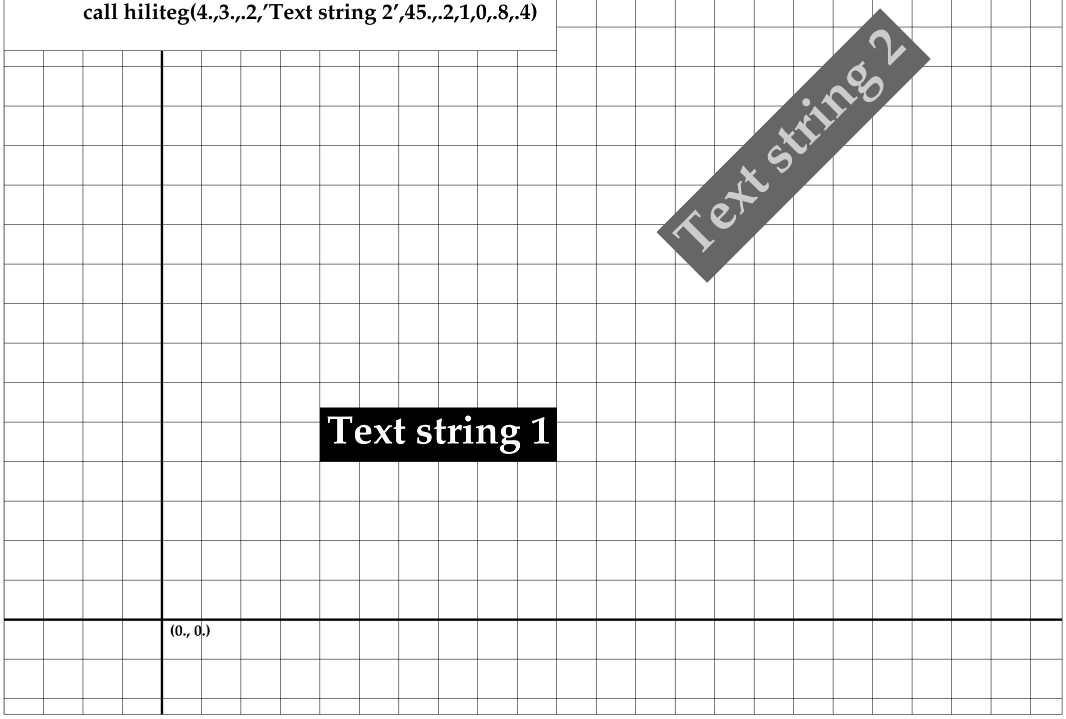
BRED, BGREEN, BBLUE Background box RGB values. Must lie between 0. and 1.0, inclusive.

See code example for HILITEG.

Example: HILITEG

Sample code:

```
call hiliteg(1.,1.,.15,'Text string 1',0.,.2,0,0,1.,0.)
call hiliteg(4.,3.,.2,'Text string 2',45.,.2,1,0,.8,.4)
```



KEKEXP

PURPOSE **KEKEXP** plots a floating point number in exponential format.

SYNTAX CALL KEKEXP (XPP, YPP, HEIGHT, FNUM, ANG, NDEC, MJUS)

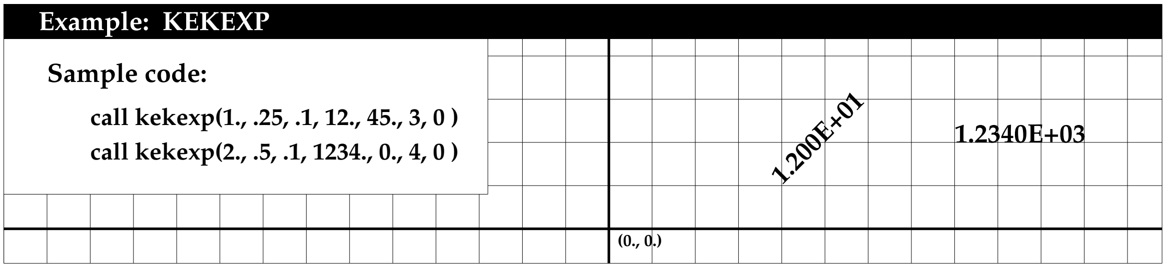
<u>*ARGUMENTS*</u>

XPP,YPP	X,Y coordinates of the number to be plotted. Plotting may be continued from the end of a previously plotted character when used in conjunction with any of the subroutines which support continuation
HEIGHT	Height of number to be plotted.
FNUM	Floating point number to be plotted.
ANG	Angle, measured counterclockwise from the X-axis, at which the number is to be plotted.
NDEC	Controls the plotted precision of FNUM. If NDEC > 0, it specifies the number of digits to the right of the decimal point, after rounding. If NDEC = 0, only the number's integer portion and a decimal point are plotted, after rounding. If NDEC = -1, only the number's integer portion is plotted, after rounding.
MJUS	Controls the justification of the number to be plotted. If MJUS = 0, (XPP,YPP) denotes the position of the lower left corner of the plotted number. If MJUS = 1, (XPP,YPP) denotes the position of the center of the plotted number. If MJUS = 2, (XPP,YPP) denotes the position of the lower right corner of the plotted number.

Example: KEKEXP

Sample code:

```
call kekexp(1., .25, .1, 12., 45., 3, 0 )
call kekexp(2., .5, .1, 1234., 0., 4, 0 )
```



KEKNUM

PURPOSE **KEKNUM** plots a floating point number in floating point format.

SYNTAX CALL KEKNUM (XPP, YPP, HEIGHT, FNUM, ANG, NDEC, MJUS)

ARGUMENTS

XPP,YPP X,Y coordinates of the number to be plotted. Plotting may be continued from the end of a previously plotted character when used in conjunction with any of the subroutines which support continuation.

HEIGHT Height of number to be plotted.

FNUM Floating point number to be plotted.

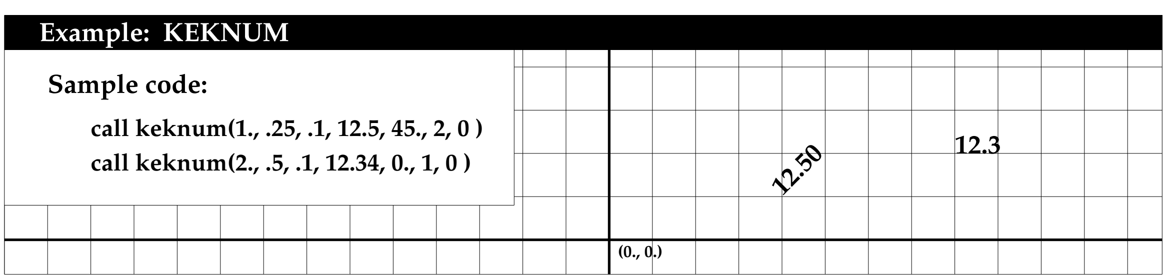
ANG Angle, measured counterclockwise from the X-axis, at which the number is to be plotted.

NDEC Controls the plotted precision of FNUM.

If NDEC > 0, it specifies the number of digits to the right of the decimal point, after rounding.
If NDEC = 0, only the number's integer portion and a decimal point are plotted, after rounding.
If NDEC = -1, only the number's integer portion is plotted, after rounding.

MJUS Controls the justification of the number to be plotted.

If MJUS = 0, (XPP,YPP) denotes the position of the lower left corner of the plotted number.
If MJUS = 1, (XPP,YPP) denotes the position of the center of the plotted number.
If MJUS = 2, (XPP,YPP) denotes the position of the lower right corner of the plotted number.

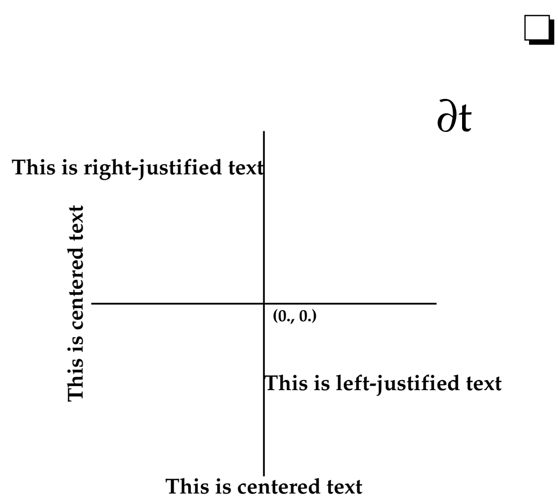


Example: KEKSYM

Sample code:

```
 call keksym(0., -1.1, .08,
1 21hThis is centered text, 0., 21, 1 )
 call keksym(-1.05, 0., .08,
1 21hThis is centered text, 90., 21, 1 )
 call keksym( 0.,-.5, .08,
1 27hThis is left-justified text, 0., 27, 0 )
 call keksym(0., .75, .08,
1 28hThis is right-justified text, 0., 28, 2 )

call setfnt(29)   !Sets font to Symbol
call keksym(1., 1., .15, 266, 0., -999, 0)
call setfnt(28)   !Sets font to Palatino-Roman
call keksym(999., 999., .15, 1ht, 0., 1, 0)
call setfnt(35)   !Sets font to Zapf Dingbats
call keksym(1.5, 1.5, .15, 157, 0., -999, 0)
```



KEKSYMO

PURPOSE **KEKSYMO** plots a character string; however, the characters are outlined rather than filled (solid).

SYNTAX CALL KEKSYMO (XPP, YPP, HEIGHT, IBCD, ANG, NCHAR, MJUS)

ARGUMENTS

XPP,YPP X,Y coordinates of the character string to be plotted. Plotting may be continued from the end of a previously plotted character when used in conjunction with any of the subroutines which support continuation.

HEIGHT Height of character string to be plotted.

IBCD Character string to be plotted (Hollerith).

ANG Angle, measured counterclockwise from the X-axis, at which the character string is to be plotted.

NCHAR Number of characters in the string to plot.

MJUS Controls the justification of the character string to be plotted.

If MJUS = 0, (XPP,YPP) denotes the position of the lower left corner of the character string.
If MJUS = 1, (XPP,YPP) denotes the position of the center of the character string.
If MJUS = 2, (XPP,YPP) denotes the position of the lower right corner of the character string.

Special feature of KEKSYMO:

You can plot special characters, for example the characters in fonts Symbol and Zapf Dingbats, by setting NCHAR=-999 and IBCD equal to the octal code of the character you want to plot.

LENSTR

PURPOSE **LENSTR** is a function whose returned value is the number of characters in the character string argument after trailing blanks have been removed.

SYNTAX LS = LENSTR(STRING, NDIM)

ARGUMENTS

STRING	Character string variable.
NDIM	Maximum length of STRING as stated in calling program.
LS	Returned length of the character string STRING after trailing blanks have been removed.

NUMBER

PURPOSE **NUMBER** is the same as KEKNUM with left justification.

SYNTAX CALL NUMBER (XPP, YPP, HEIGHT, FNUM, ANG, NDEC)

ARGUMENTS

Same as KEKNUM.

OVERBAR

PURPOSE **OVERBAR** plots a character string with an overbar.

SYNTAX CALL OVERBAR (XPP, YPP, HEIGHT, IBCD, ANG, NCHAR, MJUS)

ARGUMENTS

XPP,YPP X,Y coordinates of the character string to be plotted. Plotting may be continued from the end of a previously plotted character when used in conjunction with any of the subroutines which support continuation.

HEIGHT Height of number to be plotted.

IBCD Character string to be plotted (Hollerith).

ANG Angle, measured counterclockwise from the X-axis, at which the character string is to be plotted.

NCHAR Number of characters in the string to plot.

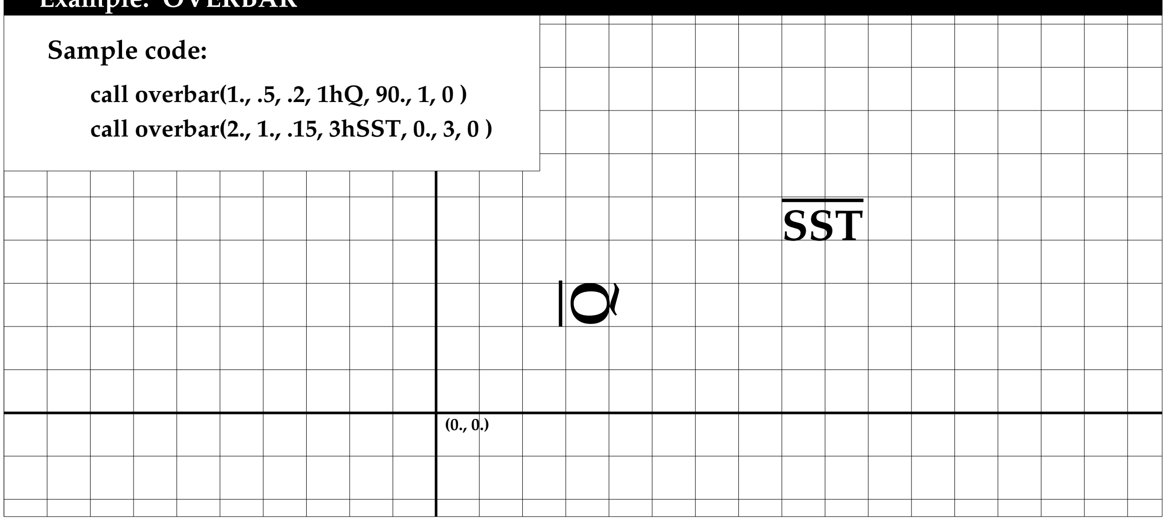
MJUS Controls the justification of the character string to be plotted.

If MJUS = 0, (XPP,YPP) denotes the position of the lower left corner of the character string.
If MJUS = 1, (XPP,YPP) denotes the position of the center of the character string.
If MJUS = 2, (XPP,YPP) denotes the position of the lower right corner of the character string.

Example: OVERBAR

Sample code:

```
call overbar(1., .5, .2, 1hQ, 90., 1, 0 )
call overbar(2., 1., .15, 3hSST, 0., 3, 0 )
```



OVERSBSPG

PURPOSE **OVERSBSPG** plots a subscripted and superscripted Greek symbol with an overbar. The overbar extends to cover both the subscripts and superscripts.

SYNTAX CALL OVERSBSPG (XPP, YPP, HEIGHT, ICH, ANG, CHAR, MJUS, ISUB, NSUB, ISUP, NSUP)

<u>*ARGUMENTS*</u>

XPP,YPP X,Y coordinates of the character string to be plotted. Plotting may be continued from the end of a previously plotted character when used in conjunction with any of the subroutines which support continuation.

HEIGHT Height of symbol to be plotted.

ICH Integer value between 1 and 50 corresponding to the desired Greek symbol (see table below Subroutine GRKSYM).

ANG Angle, measured counterclockwise from the X-axis, at which the character string is to be plotted.

NCHAR Number of characters in the string to plot (should be set to 1).

MJUS Controls the justification of the character string to be plotted.

If MJUS = 0, (XPP,YPP) denotes the position of the lower left corner of the character string.
If MJUS = 1, (XPP,YPP) denotes the position of the center of the character string.
If MJUS = 2, (XPP,YPP) denotes the position of the lower right corner of the character string.

ISUB Hollerith character(s) comprising the subscript.

NSUB Number of characters in the subscript string.

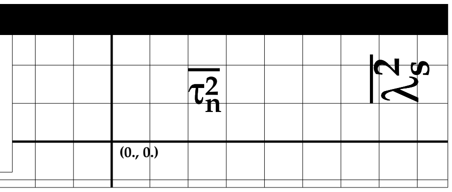
ISUP Hollerith character(s) comprising the superscript.

NSUP Number of characters in the superscript string.

Example: OVERSBSPG

Sample code:

```
call oversbspg(2., .25, .2, 35, 90., 1, 0, 1hs, 1, 1h2, 1 )
call oversbspg(.5, .25, .15, 43, 0., 1, 0, 1hn, 1, 1h2, 1 )
```



OVERSUBG

PURPOSE **OVERSUBG** plots a subscripted Greek symbol with an overbar. The overbar extends to cover the subscript.

SYNTAX CALL OVERSUBG(XPP, YPP, HEIGHT, ICH, ANG, NCHAR, MJUS,ISUB, NSUB)

ARGUMENTS

XPP,YPP X,Y coordinates of the character string to be plotted. Plotting may be continued from the end of a previously plotted character when used in conjunction with any of the subroutines which support continuation.

HEIGHT Height of the Greek symbol to be plotted.

ICH Integer value between 1 and 50 corresponding to the desired Greek symbol (see table below Subroutine GRKSYM).

ANG Angle, measured counterclockwise from the X-axis, at which the symbol is to be plotted.

NCHAR Number of characters in the string to plot (should be set to 1).

MJUS Controls the justification of the character string to be plotted.

If MJUS = 0, (XPP,YPP) denotes the position of the lower left corner of the symbol.
If MJUS = 1, (XPP,YPP) denotes the position of the center of the symbol.
If MJUS = 2, (XPP,YPP) denotes the position of the lower right corner of the symbol.

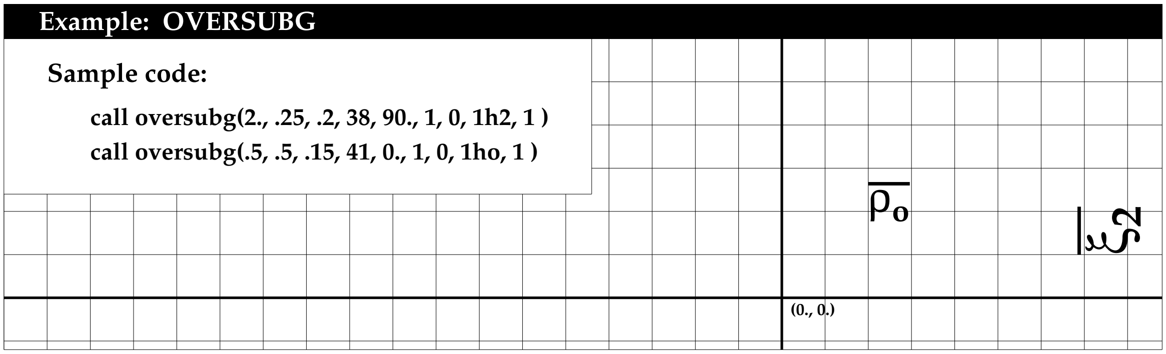
ISUB Hollerith character(s) comprising the subscript.

NSUB Number of characters in the subscript string.

Example: OVERSUBG

Sample code:

```
call oversubg(2., .25, .2, 38, 90., 1, 0, 1h2, 1 )
call oversubg(.5, .5, .15, 41, 0., 1, 0, 1ho, 1 )
```



OVERSUPG

PURPOSE **OVERSUPG** plots a subscripted Greek character with an overbar. The overbar extends to cover the superscript.

SYNTAX CALL OVERSUPG(XPP, YPP, HEIGHT, ICH, ANG, NCHAR, MJUS, ISUP, NSUP)

ARGUMENTS

XPP,YPP X,Y coordinates of the Greek symbol to be plotted. Plotting may be continued from the end of a previously plotted character when used in conjunction with any of the subroutines which support continuation.

HEIGHT Height of the Greek symbol to be plotted.

ICH Integer value between 1 and 50 corresponding to the desired Greek symbol (see table below Subroutine GRKSYM).

ANG Angle, measured counterclockwise from the X-axis, at which the symbol is to be plotted.

NCHAR Number of characters in the string to plot (should be set to 1).

MJUS Controls the justification of the character string to be plotted.

If MJUS = 0, (XPP,YPP) denotes the position of the lower left corner of the symbol.
If MJUS = 1, (XPP,YPP) denotes the position of the center of the symbol.
If MJUS = 2, (XPP,YPP) denotes the position of the lower right corner of the symbol.

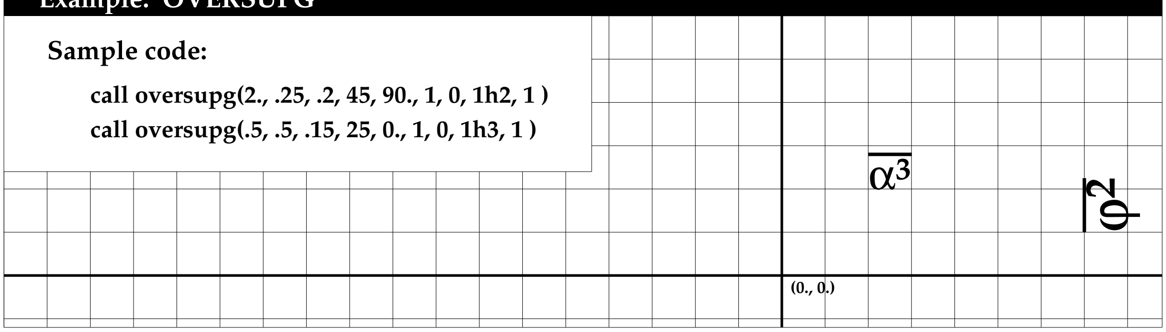
ISUP Hollerith character(s) comprising the superscript.

NSUP Number of characters in the superscript string.

Example: OVERSUPG

Sample code:

```
call oversupg(2., .25, .2, 45, 90., 1, 0, 1h2, 1 )
call oversupg(.5, .5, .15, 25, 0., 1, 0, 1h3, 1 )
```



PLOT

PURPOSE **PLOT** is the most fundamental user-level plotting command. It gives you direct control of pen movement (to any X,Y coordinate position) and pen status (up or down). Additionally, it allows you to re-define the current plotting origin.

SYNTAX CALL PLOT (XPP, YPP, IPEN)

ARGUMENTS

XPP,YPP X,Y coordinates of the position to which the pen is to be moved. An origin may be established anywhere on or off the plotting surface, as explained below for negative IPEN values.

IPEN A signed integer which controls pen status (up/down) and origin definition.

If IPEN = 2, the pen is down during movement, thus drawing a visible line.

If IPEN = 3, the pen is up during movement, thus changing the pen's current position only.

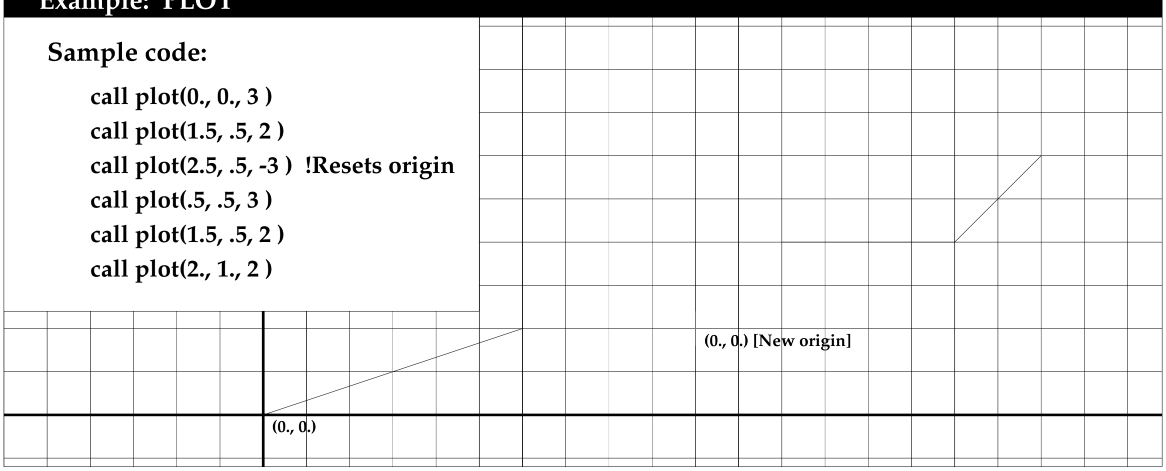
If IPEN = -2 or -3, a new origin is defined at the position (XPP,YPP) after the movement is completed as if IPEN were positive. The logical X,Y coordinates of the new pen position are set to (0,0), so that all subsequent pen movements use this position as a reference point.

If IPEN = 999, the call to PLOT closes the output file. Thus, a call to PLOT with IPEN = 999 may be used only once in a given plotting session, and if used, must be the last plotting command in the plotting session. Calling PLOT with IPEN = 999 is identical to calling PLOTND (which see).

Example: PLOT

Sample code:

```
call plot(0., 0., 3 )
call plot(1.5, .5, 2 )
call plot(2.5, .5, -3 )  !Resets origin
call plot(.5, .5, 3 )
call plot(1.5, .5, 2 )
call plot(2., 1., 2 )
```



PLSMIN

PURPOSE **PLSMIN** plots the symbol ±.

SYNTAX CALL PLSMIN (XPP, YPP, HEIGHT, ANG, MJUS)

ARGUMENTS

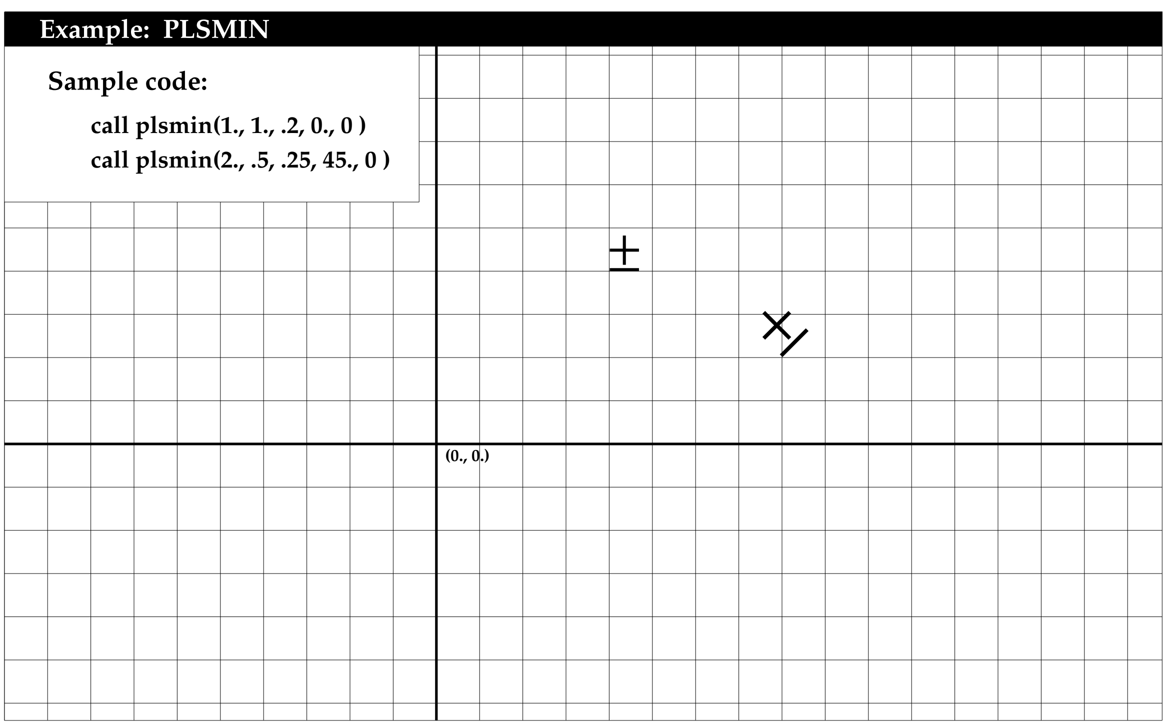
XPP,YPP X,Y coordinates of the symbol to be plotted. Plotting may be continued from the end of a previously plotted character when used in conjunction with any of the subroutines which support continuation.

HEIGHT Height of symbol to be plotted.

ANG Angle, measured counterclockwise from the X-axis, at which the symbol is to be plotted.

MJUS Controls the justification of the character string to be plotted.

If MJUS = 0, (XPP,YPP) denotes the position of the lower left corner of the character.
If MJUS = 1, (XPP,YPP) denotes the position of the center of the character.
If MJUS = 2, (XPP,YPP) denotes the position of the lower right corner of the character.



PSINIT

PURPOSE **PSINIT** is called to begin a plotting session. It must be called before any other plotting command, with the exception of NEWDEV.

SYNTAX CALL PSINIT (PRTRT)

ARGUMENTS

PRTRT Logical variable indicating the paper orientation. If PRTRT=.TRUE., the paper is oriented in portrait mode, i.e. long side of page is vertical. If PRTRT=.false., the paper is oriented in landscape mode, i.e. long side of page is horizontal.

RECTFILC

PURPOSE **RECTFILC** draws a rectangle (or square) and fills it with a specified red, green, and blue color levels.

SYNTAX CALL RECTFILC(XX1, YY1, XX2, YY2, HEIGHT, RED, GREEN BLUE)

ARGUMENTS

XX1,YY1 X,Y coordinates of lower left corner of rectangle.

XX2,YY2 X,Y coordinates of lower right corner of rectangle.

HEIGHT Height of rectangle.

RED, GREEN, BLUE RGB values of the fill color. Must lie between 0. and 1., inclusively.

ROTATE

PURPOSE **ROTATE** rotates the current coordinate system by a specified angle. Essentially, ROTATE issues the PostScript *rotate* command.

SYNTAX CALL ROTATE (ANG)

ARGUMENTS

ANGLE Angle, measured counterclockwise, to rotate the current coordinate system.

SETCOLR

PURPOSE **SETCOLR** sets the current red, green, and blue color levels.

SYNTAX CALL SETCOLR (RED, GREEN, BLUE)

ARGUMENTS

RED	Red color saturation level, between 0. and 1.0.
GREEN	Green color saturation level, between 0. and 1.0.
BLUE	Blue color saturation level, between 0. and 1.0.

Characters and octal codes for Font Symbol

	!	∀	#	∃	%	&	∋	(	)	∗	+	,	−	.	/
40	**41**	**42**	**43**	**44**	**45**	**46**	**47**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
60	**61**	**62**	**63**	**64**	**65**	**66**	**67**	**70**	**71**	**72**	**73**	**74**	**75**	**76**	**77**
≅	Α	Β	Χ	Δ	Ε	Φ	Γ	Η	Ι	ϑ	Κ	Λ	Μ	Ν	Ο
100	**101**	**102**	**103**	**104**	**105**	**106**	**107**	**110**	**111**	**112**	**113**	**114**	**115**	**116**	**117**
Π	Θ	Ρ	Σ	Τ	Υ	ς	Ω	Ξ	Ψ	Ζ	[	∴	]	⊥	_
120	**121**	**122**	**123**	**124**	**125**	**126**	**127**	**130**	**131**	**132**	**133**	**134**	**135**	**136**	**137**
‾	α	β	χ	δ	ε	ϕ	γ	η	ι	φ	κ	λ	μ	ν	ο
140	**141**	**142**	**143**	**144**	**145**	**146**	**147**	**150**	**151**	**152**	**153**	**154**	**155**	**156**	**157**
π	θ	ρ	σ	τ	υ	ϖ	ω	ξ	ψ	ζ	{	|	}	∼	
160	**161**	**162**	**163**	**164**	**165**	**166**	**167**	**170**	**171**	**172**	**173**	**174**	**175**	**176**	**177**
€	ϒ	′	≤	⁄	∞	ƒ	♣	♦	♥	♠	↔	←	↑	→	↓
240	**241**	**242**	**243**	**244**	**245**	**246**	**247**	**250**	**251**	**252**	**253**	**254**	**255**	**256**	**257**
°	±	″	≥	×	∝	∂	•	÷	≠	≡	≈	…	⏐	⎯	↵
260	**261**	**262**	**263**	**264**	**265**	**266**	**267**	**270**	**271**	**272**	**273**	**274**	**275**	**276**	**277**
ℵ	ℑ	ℜ	℘	⊗	⊕	∅	∩	∪	⊃	⊇	⊄	⊂	⊆	∈	∉
300	**301**	**302**	**303**	**304**	**305**	**306**	**307**	**310**	**311**	**312**	**313**	**314**	**315**	**316**	**317**
∠	∇	®	©	™	∏	√	⋅	¬	∧	∨	⇔	⇐	⇑	⇒	⇓
320	**321**	**322**	**323**	**324**	**325**	**326**	**327**	**330**	**331**	**332**	**333**	**334**	**335**	**336**	**337**
◊	〈	®	©	™	∑	⎛	⎜	⎝	⎡	⎢	⎣	⎧	⎨	⎩	⎪
340	**341**	**342**	**343**	**344**	**345**	**346**	**347**	**350**	**351**	**352**	**353**	**354**	**355**	**356**	**357**
	〉	∫	⌠	⎮	⌡	⎞	⎟	⎠	⎤	⎥	⎦	⎫	⎬	⎭	
360	**361**	**362**	**363**	**364**	**365**	**366**	**367**	**370**	**371**	**372**	**373**	**374**	**375**	**376**	

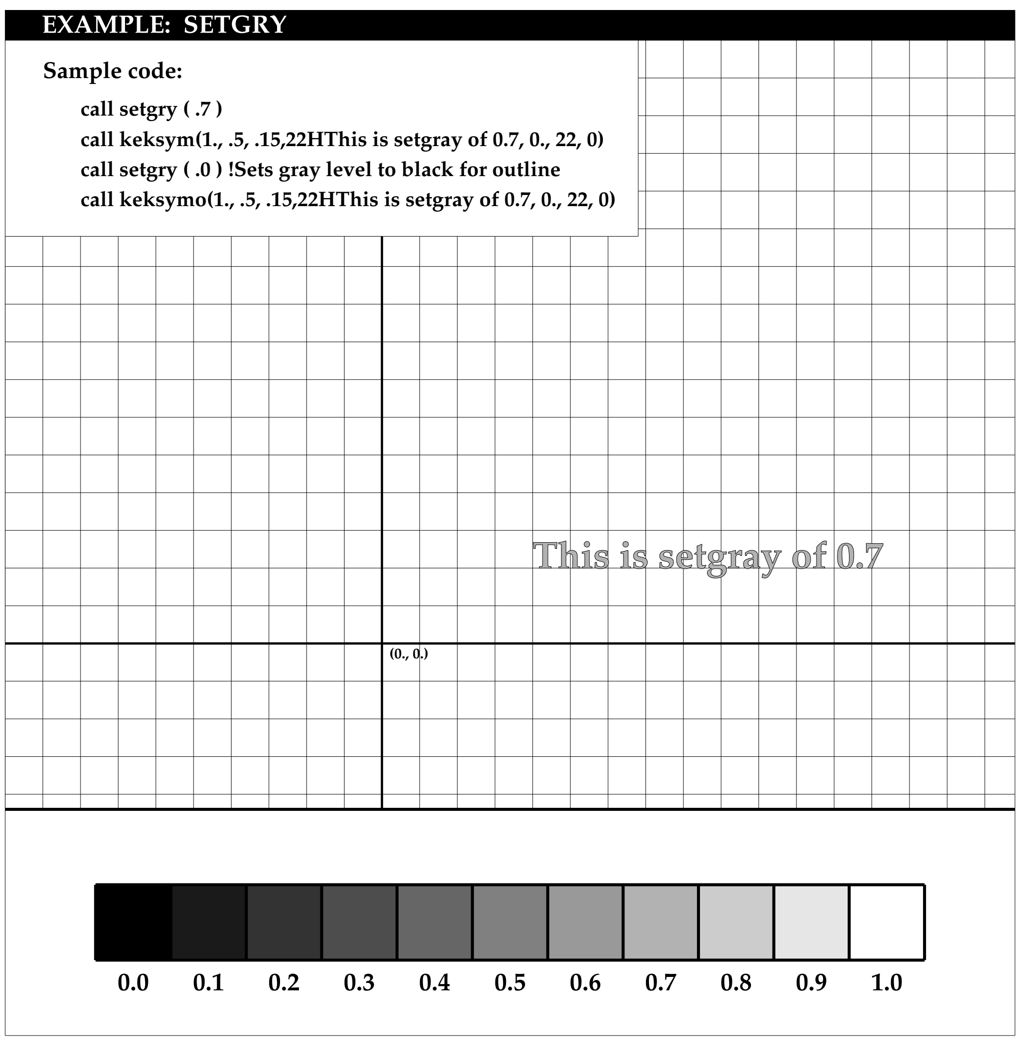
SETGRY

PURPOSE **SETGRY** sets the current gray level.

SYNTAX CALL SETGRY (GRYLVL)

ARGUMENTS

GRYLVL Gray level between 0. (black) and 1. (white).



SIGMA

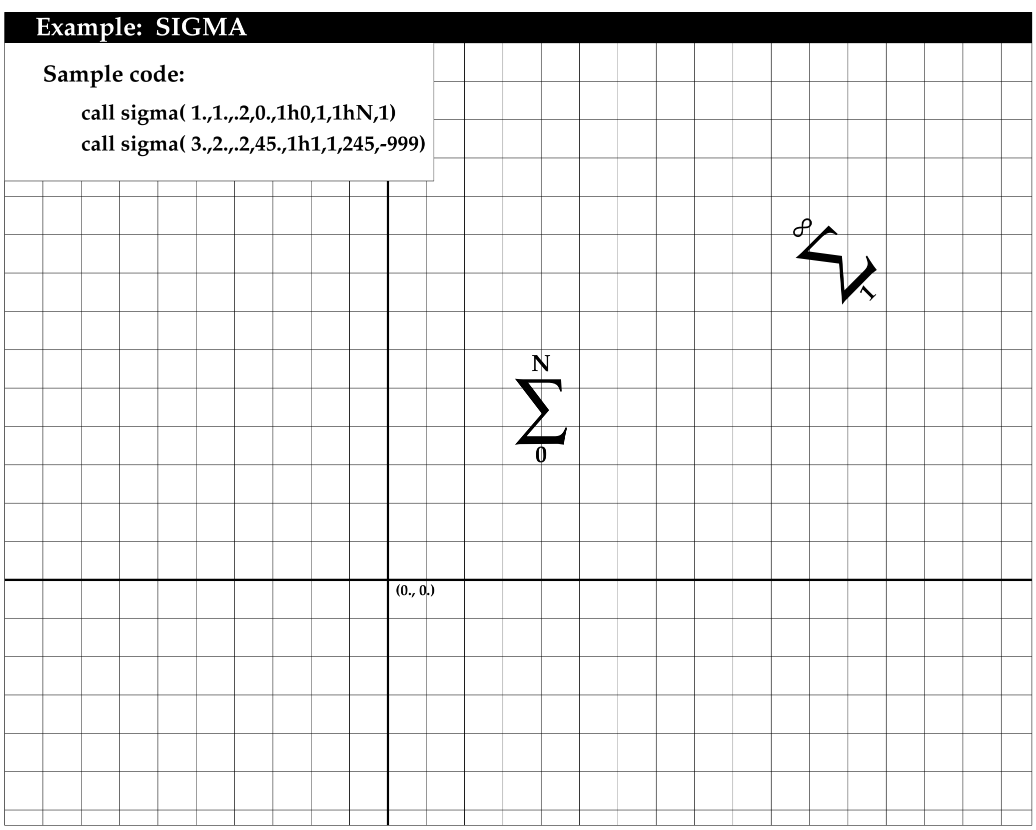
PURPOSE **SIGMA** draws the symbol Σ, with upper and lower limits.

SYNTAX CALL SIGMA (XPP, YPP, HEIGHT, ANG, LOWER, NL, LUPPER, NU)

ARGUMENTS

XPP, YPP	X, Y coordinates of the lower left edge of the symbol.
HEIGHT	Height of the symbol.
ANG	Angle, measured counterclockwise from the X-axis, at which the symbol is to be plotted.
LOWER	Character string specifying the lower limit (Hollerith).
NL	Number of characters in the lower limit character string.
LUPPER	Character string specifying the upper limit (Hollerith).
NU	Number of characters in the upper limit character string.

If NL (NU) is set to -999, then LOWER (LUPPER) is the octal code of the character in the font SYMBOL.



SLDLIN

PURPOSE **SLDLIN** connects two points using a line with a specified thickness.

SYNTAX CALL SLDLIN (X1, Y1, X2, Y2, THK)

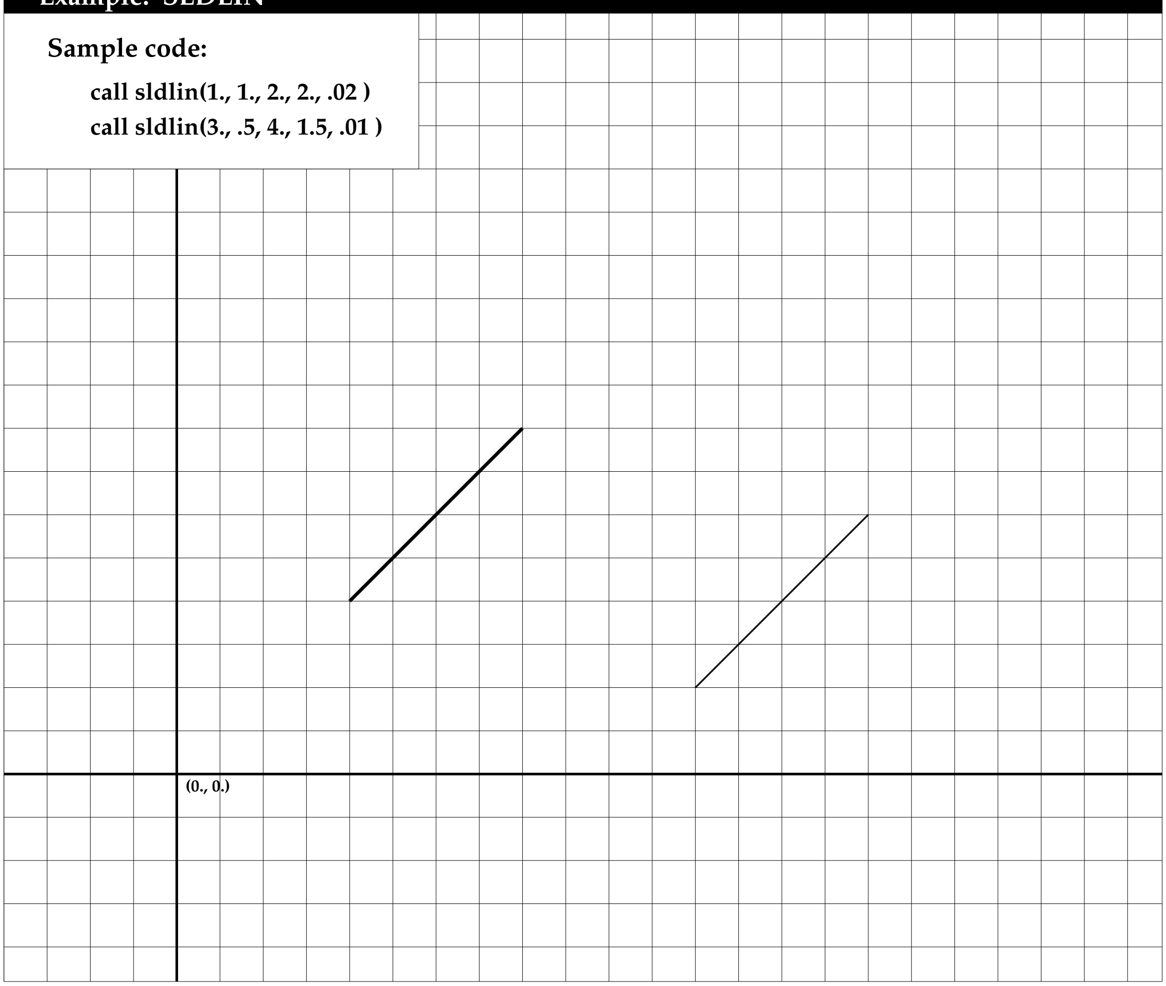
ARGUMENTS

X1,Y1	X,Y coordinates of first data point.
X2,Y2	X,Y coordinates of second data point.
THK	Specifies the thickness of the connecting line. If THK=0., the current linewidth is used.

Example: SLDLIN

Sample code:

```
call sldlin(1., 1., 2., 2., .02 )
call sldlin(3., .5, 4., 1.5, .01 )
```



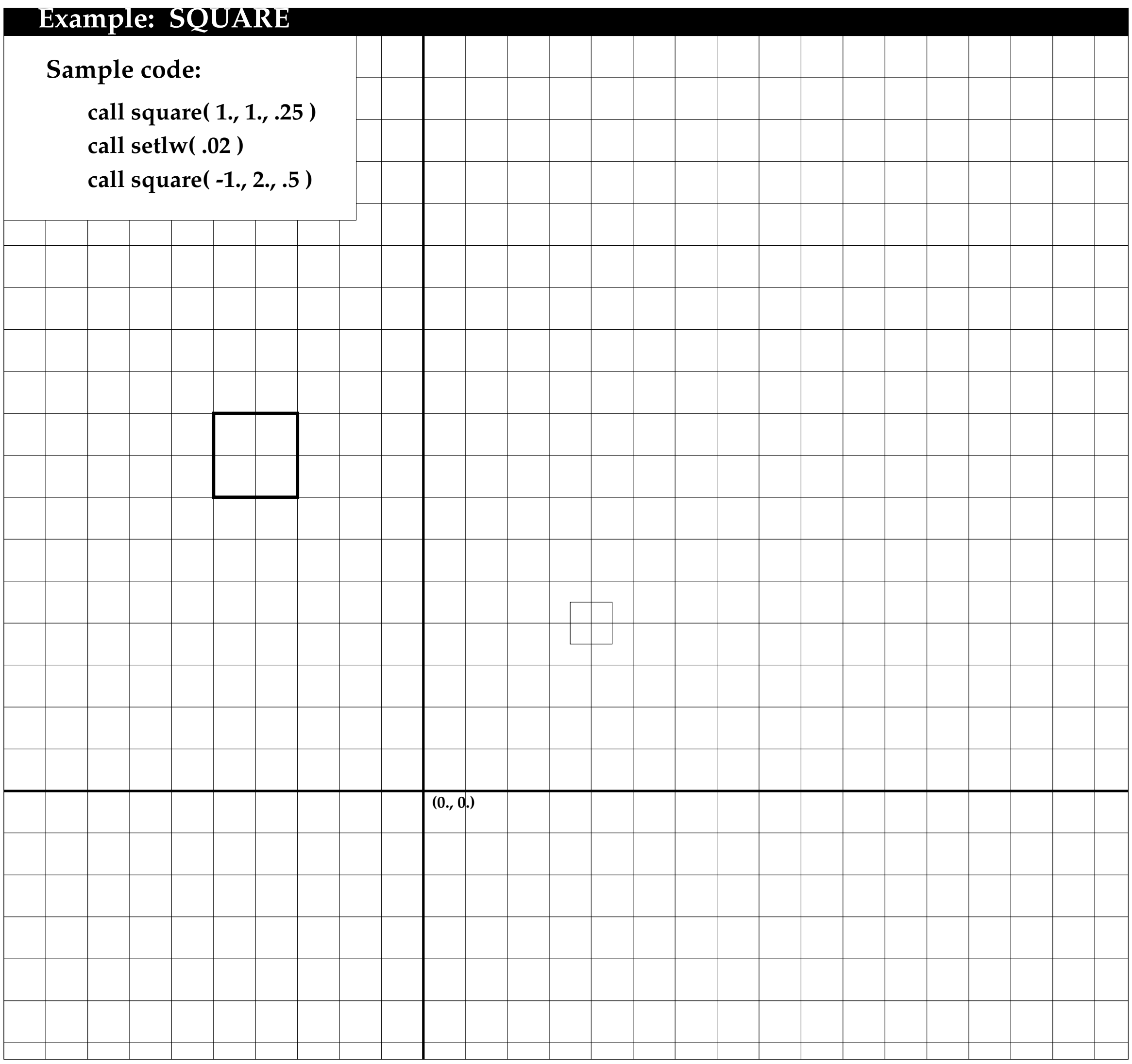
SQUARE

PURPOSE **SQUARE** draws a centered square.

SYNTAX CALL SQUARE (XC, YC, SIDE)

ARGUMENTS

XC,YC	X,Y coordinates of the center of the square.
SIDE	Length of the sides of the square.



SUBBER

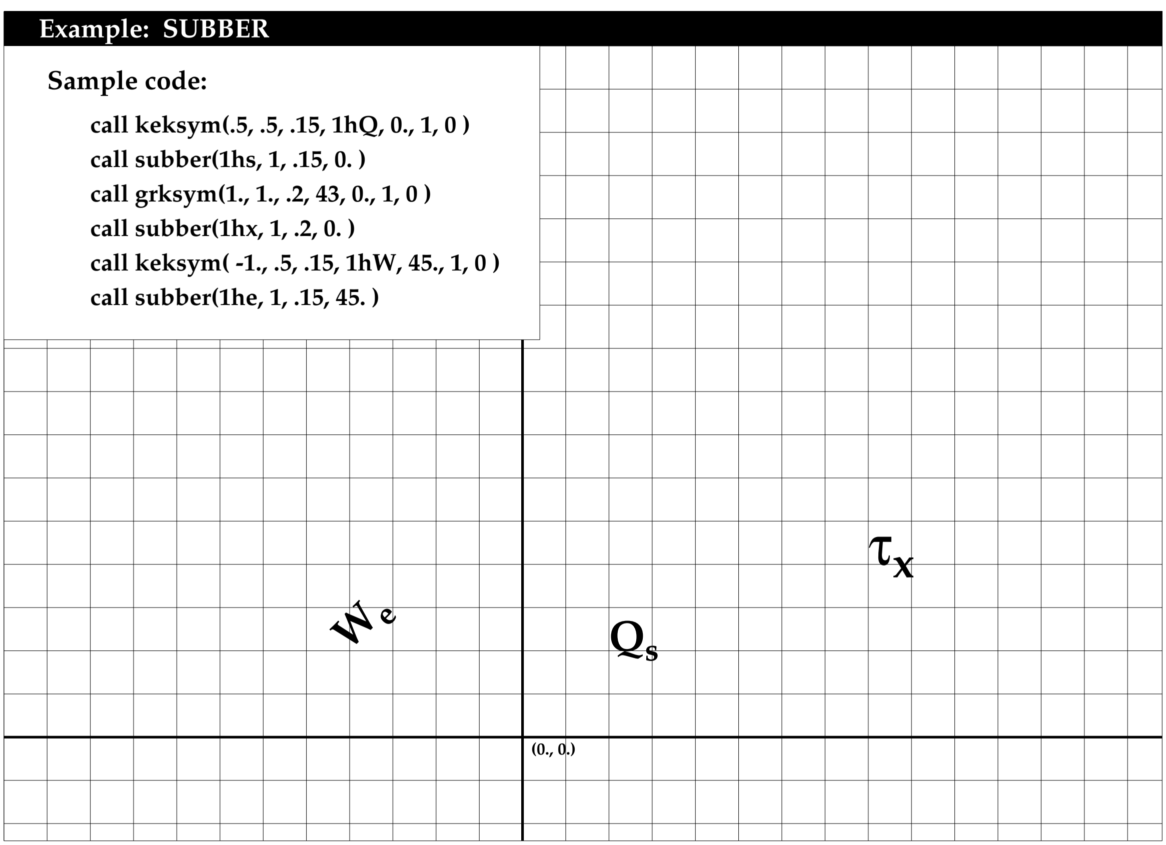
PURPOSE **SUBBER** draws a subscript.

SYNTAX CALL SUBBER (ISUB, NSUB, SIZE, ANG)

ARGUMENTS

ISUB	Hollerith variable specifying the subscript character string.
NSUB	Number of characters in the subscript.
SIZE	Height of the **variable to be subscripted**. It is **not** the height of the subscript characters themselves.
ANG	Angle of the subscripted variable.

Note: SUBBER must be called immediately after the call to create the subscripted variable.



SUBSUP

PURPOSE **SUBSUP** draws a subscript and superscript.

SYNTAX CALL SUBSUP (ISUB, NSUB, ISUP, NSUP, SIZE, ANG)

ARGUMENTS

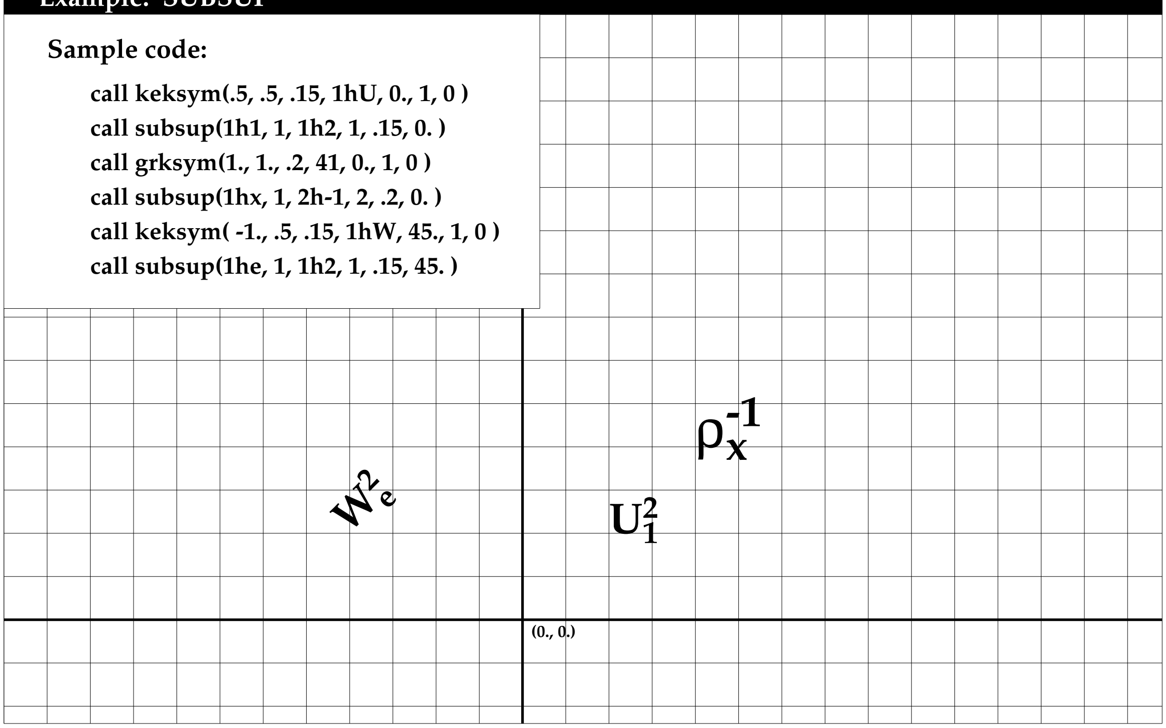
ISUB	Hollerith variable specifying the subscript character string.
NSUB	Number of characters in the subscript.
ISUP	Hollerith variable specifying the superscript character string.
NSUP	Number of characters in the superscript.
SIZE	Height of the **variable to be subscripted and superscripted**. It is **not** the height of the subscript and superscript characters themselves.
ANG	Angle of the subscripted and superscripted variable.

Note: SUBSUP must be called immediately after the call to create the subscripted and superscripted variable.

Example: SUBSUP

Sample code:

```
call keksym(.5, .5, .15, 1hU, 0., 1, 0 )
call subsup(1h1, 1, 1h2, 1, .15, 0. )
call grksym(1., 1., .2, 41, 0., 1, 0 )
call subsup(1hx, 1, 2h-1, 2, .2, 0. )
call keksym( -1., .5, .15, 1hW, 45., 1, 0 )
call subsup(1he, 1, 1h2, 1, .15, 45. )
```



SUPER

PURPOSE **SUPER** draws a superscript.

SYNTAX CALL SUPER (ISUP, NSUP, SIZE, ANG)

ARGUMENTS

ISUP Hollerith variable specifying the superscript character string.

NSUP Number of characters in the superscript.

SIZE Height of the **variable to be superscripted**. It is **not** the height of the subscript characters themselves.

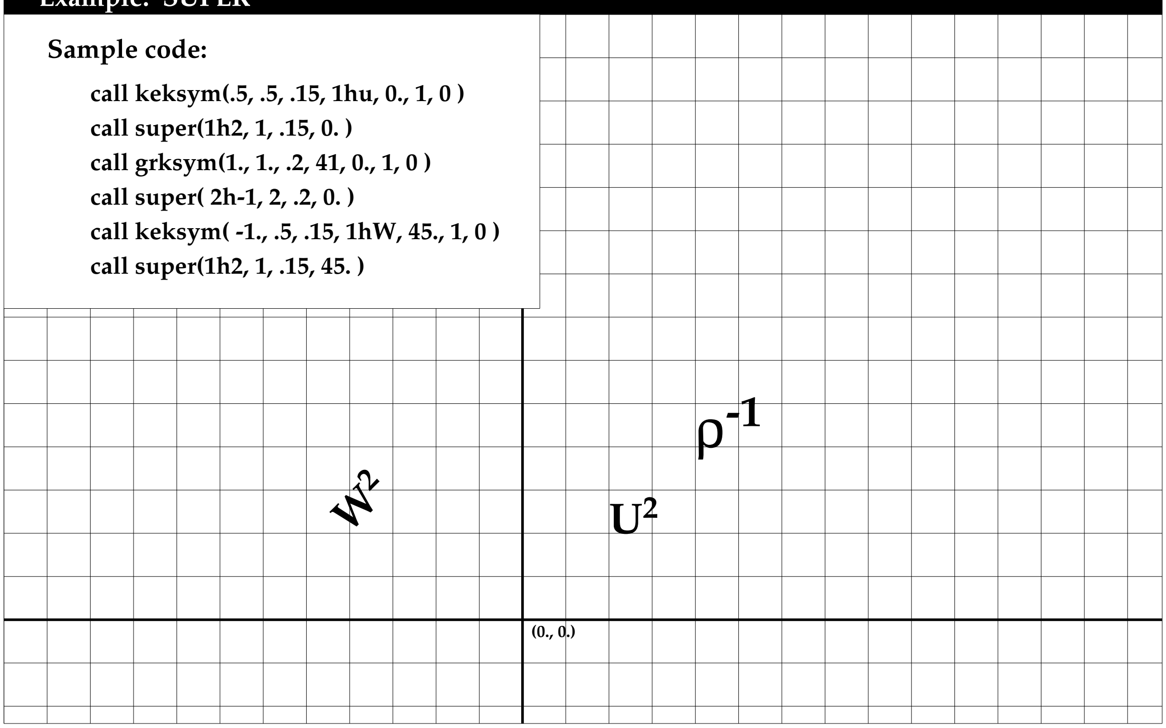
ANG Angle of the superscripted variable.

Note: SUPER must be called immediately after the call to create the superscripted variable.

Example: SUPER

Sample code:

```
call keksym(.5, .5, .15, 1hu, 0., 1, 0 )
call super(1h2, 1, .15, 0. )
call grksym(1., 1., .2, 41, 0., 1, 0 )
call super( 2h-1, 2, .2, 0. )
call keksym( -1., .5, .15, 1hW, 45., 1, 0 )
call super(1h2, 1, .15, 45. )
```



SYMBOL

PURPOSE **SYMBOL** plots a character string. It is similar to subroutine KEKSYM, except that justification and continuation are not supported.

SYNTAX CALL SYMBOL (XPP, YPP, HEIGHT, IBCD, ANG, NCHAR)

ARGUMENTS

XPP,YPP X,Y coordinates of the lower left corner of the first character of the string to be plotted.

HEIGHT Height of character string to be plotted.

IBCD Character string to be plotted (Hollerith).

ANG Angle, measured counterclockwise from the X-axis, at which the character string is to be plotted.

NCHAR Number of characters in the string.

Example: SYMBOL

Sample code:

```
call symbol( -1.5, .5, .15, 8hVelocity, 0., 8 )
call symbol(1.5,.5, .15, 8hDistance, 45., 8 )
```