Data Step Trigonometry for the Universal NO Symbol

Kim J. Kubasek, K Info Management, Arroyo Grande, CA

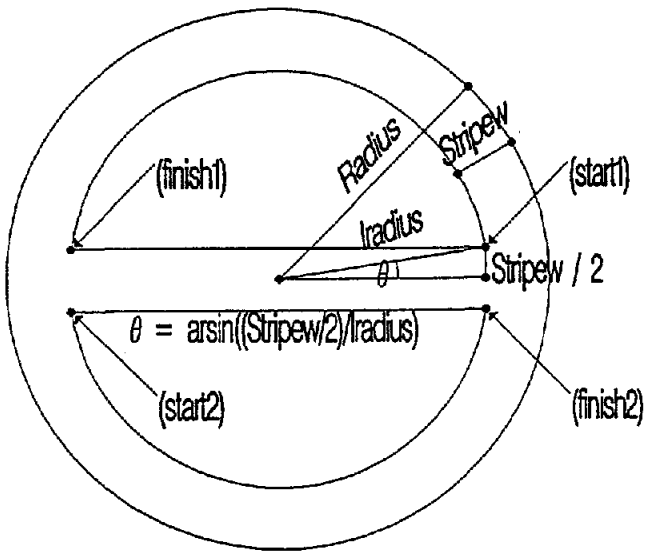
ABSTRACT

Using parameters such as radius, stripe width and stripe angle, a DATA step can be written to generate the X,Y coordinate pairs which describe the Universal NO Symbol (as in BOZOS or GHOSTS). These coordinates can be used in an annotate DATA set or to create a font or a map.

INTRODUCTION

In preparation for the SUGI '88 Conference, BASAS decided to design a shirt with a logo (see first example). I volunteered to design the logo using SAS/GRAPH® software. The ANNOTATE facility of the SAS® System was chosen so that a flexible DATA step could be written in which various aspects of the symbol could be manipulated. As the code "grew" it became a study in the use of various aspects of the trigonometry functions in the DATA step. The resulting DATA set turned out to be equally functional as an ANNOTATE DATA set, a GFONT DATA set, or GMAP DATA set.

Trigonometric Features of the NO Symbol



BASIC TRIGONOMETRY

The outer circle of the symbol can be considered as the unit circle of trigonometry. The center of the symbol is the origin (x=0, y=0) of the unit circle. The distance from the center to the outer edge of is the radius. For clarity, imagine the symbol with the stripe going horizontally (a stripe angle of zero). Now imagine that the symbol is formed by removing two partial half-circles from a solid circle.

The beginning of the top part to be removed is labeled as (start1). It is located at the end of the inside radius (**Iradius**) at an angle which will leave half a stripe width(**Stripew**) between it and the center line of the symbol. This angle is calculated as:

$$\text{Start Angle} = \arcsin\left(\frac{(\text{Stripe Width})/2}{(\text{Inside radius})}\right)$$

$$\text{Finish Angle} = \pi - \text{Start Angle}$$

If we add to the start and finish angle some additional amount (say 45°) the stripe can be moved to a more typical angle for the symbol. If an arc is drawn using the inside radius from this angle to its reflected angle on the left side of the symbol (finish1) and then a straight line back to (start1), the top part is complete.

This arc can be approximated with a polygon created in a "do-loop" by iterating an angle (A) from the start angle to the finish angle by some small amount (delta) and calculating the location of the points as:

$$X = \cos(A) * (\text{Inside radius})$$

$$Y = \sin(A) * (\text{Inside radius})$$

If a similar figure is drawn as the bottom part to be removed, then what is left is the NO symbol.

SPECIAL CONSIDERATIONS

1) The annotate facility uses degrees to specify angles. The DATA step functions use radians to specify angles. There is a simple conversion based on two pi radians in a circle versus 360 degrees in a circle. Thus:

$$\text{Radians} = \frac{(\pi * \text{Degrees})}{180}$$

A handy way to calculate pi is:

$$\pi = \arccos(-1)$$

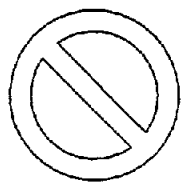
2) To indicate polygons within a figure that are not part of that figure, the SAS DATA set should contain an observation with missing values for X and Y between each polygon.

3) When iterating a do loop by some non-integer value, sometimes the last iteration will be dropped because of round-off error. To circumvent this problem subtract a small amount from the last value, and then specify the last value after a comma. For example from the actual code:

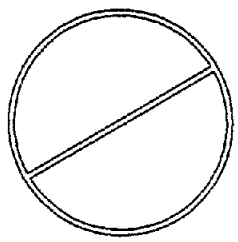
```
** adjust for aspect ratio ;
if ratio > 1 then do ;  x_mult=1 ;  y_mult=1/ratio ;
  end ;
else do ;  x_mult=ratio ;  y_mult=1 ;
  end ;

do a=start1 to finish1-delta/2 by delta , finish1 ;
  x=cx+iradius*x_mult*cos(a) ;
  y=cy+iradius*y_mult*sin(a) ;
  output ;
end
```

EXAMPLES:

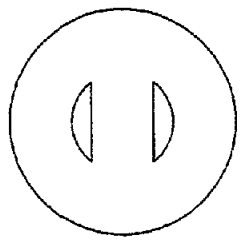
Stripe Radius=25%
Stripe Width=3%
Angle=45 degrees



Stripe Radius=15%
Stripe Width=3.75%
Angle=135 degrees



Stripe Radius=20%
Stripe Width=1%
Angle=30 degrees



Stripe Radius=20%
Stripe Width=11%
Angle=90 degrees

```
/* NOWUSS9e.SAS                                                    */
/* Code to generate the "No SASHOLES" symbol for BASAS             */
/* Demonstrates use of trigonometric functions to draw symbols     */
/* as well as some techniques for various graphics procedures.     */

/* Kim J. Kubasek    7/13/94                                       */

%macro makeno(pattern=empty, color=red, stripea=45, stripew=8,
              xcenter=50, ycenter=50, radius=50, ratio=1) ;

* ratio from gtestit ;
* ratio=0.78750 ;  * hpljs2 device landscape  ;
* ratio=1.26984 ;  * hpljs2 device portrait   ;
* ratio=0.58836 ;  * gdevice=WIN              ;
* ratio=0.58118 ;  * gdevice=OS2              ;
* ratio=1 ;        * use for fonts and maps   ;
* use this ratio to:           ;
* multiply to X offset (from center) if less than one ;
* multiply inverse to Y offset if greater than one    ;
* may take some trial and error adjustment for each   ;
* device to make a round circle                       ;

ratio=&ratio ;
* calculate multipliers for X and Y ;
if ratio > 1 then do ;  x_mult=1 ;  y_mult=1/ratio ;
  end ;
else do ;  x_mult=ratio ;  y_mult=1 ;
  end ;

pi=arcos(-1) ;           * 2 pi radians in a circle ;
delta=pi/50  ;           * resolution of arcs -- 100 per circle ;
cx = &xcenter ;          * center x coordinate  in % ;
cy = &ycenter ;          * center y coordinate  in % ;
radius=&radius;          * outside radius in pct -- max of 50 ;
stripea = &stripea*pi/180 ;  * convert angle to radians ;

stripew=&stripew ;       * stripe width in pct:  max = 2/3 radius ;
iradius=radius-stripew ; * inside radius in pct ;

* calculate radians of 1/2 of a stripe width = theta ;
sin=(stripew/2)/iradius;
theta=arsin(sin) ;
start1 = stripea+theta ;  * area on one side of stripe ;
finish1 = pi+stripea-theta ;

start2 = start1 + pi ;       * area on other side ;
finish2 = finish1 + pi ;

retain xsys ysys hsys '3' ;  * percent of screen ;
length color function style $ 8 when $ 1 text $ 200 ;

*************************************************************;
*   variables needed for a font definition                  ;
*************************************************************;
retain lp 'p' char 'N' ptype 'v' fill 's' segment 1 ;

*************************************************************;
* now start to draw a "polygon" which is the "NO" symbol    ;
*************************************************************;

*************************************************************;
* first set a few parameters for the symbol                 ;
*************************************************************;
size=1           ;   * thickness of outline                  ;
line=1           ;   * line type of outline                  ;
when='A'         ;   * draw symbol on top of (After) text    ;
color="&color"   ;   * color of body of symbol              ;
function='poly'  ;   * to begin a polygon                    ;
style="&pattern" ;   * not quite solid -- displays well      ;

*************************************************************;
* next, draw the outsize of the circle                      ;
*************************************************************;
do a=0 to 2*pi-delta/2  by delta ;
  x=cx+radius*x_mult*cos(a) ;
  y=cy+radius*y_mult*sin(a) ;
  output ;
  function='polycont' ;
  color='black  ' ;  * color of the outline of the symbol ;
  end ;

*************************************************************;
* an observation to separate parts of the polygon           ;
*************************************************************;
x=. ;
y=. ;
output ;

*************************************************************;
* now the top half of the inside of the symbol              ;
*************************************************************;
do a=start1 to finish1-delta/2 by delta , finish1 ;
  x=cx+iradius*x_mult*cos(a) ;
  y=cy+iradius*y_mult*sin(a) ;
  output ;
  end ;

*************************************************************;
* another observation to separate parts of the polygon      ;
*************************************************************;
x=. ;
y=. ;
output ;

*************************************************************;
* finally, the bottom half of the inside of the symbol      ;
*************************************************************;
do a=start2 to finish2 -delta/2 by delta , finish2 ;
  x=cx+iradius*x_mult*cos(a) ;
  y=cy+iradius*y_mult*sin(a) ;
  output ;
  end ;

color='black' ;
function='label' ;
style='swissl' ;
size=2.8 ;
position='5' ;
x=cx ;
y=cy - y_mult*(radius+4) ;
text="Stripe Radius=&radius%" ;
output ;

y=y-3.1 ;
text="Stripe Width=&stripew%" ;
output ;

y=y-3.1 ;
text="Angle=&stripea degrees" ;
output ;

%mend ;

options mprint ;

%let vsize=23 ;
%let hsize=18 ;
%global vorigin ;

data one ;
%makeno(pattern=S     , color=black  , stripea=45, stripew=3,
      xcenter=30, ycenter=79, radius=25,  ratio=&vsize/&hsize ) ;
%makeno(pattern=empty,  color=red,  stripea=135, stripew=3.75,
      xcenter=77.5, ycenter=84, radius=15,  ratio=&vsize/&hsize ) ;
%makeno(pattern=empty,  color=red,  stripea=30, stripew=1,
      xcenter=25, ycenter=28, radius=20,  ratio=&vsize/&hsize ) ;
%makeno(pattern=empty,  color=red,  stripea=90, stripew=11,
      xcenter=75, ycenter=28, radius=20,  ratio=&vsize/&hsize ) ;

x=30 ;
y=80 ;
color='black' ;
size=6 ;
style='swissb' ;
function='label' ;
position='5' ;
text='SASHOLES' ;
when='B' ;
output ;

run ;

* display instructions with GANNO ;

goptions reset=all targetdevice=HPGL
 vsize=23 cm hsize=18 cm horigin= 1 cm vorigin=1 cm
 prompt display gsfname=HGL gsfmode=REPLACE gsflen=250
 ;

filename HGL 'nowuss9e.HGL' ;

proc ganno anno=one ;
run ;
```

Requests for e-mail copies of the code may be sent to the author at 71561.3114@CompuServe.COM