Subject: Re: need more plotting symbols please
Posted by David Fanning on Thu, 31 Aug 2006 23:14:51 GMT
View Forum Message <> Reply to Message

Dilkushi@gmail.com writes:

> In my assignment the number of variables I am requested to plot are
> increasing (exponentially !!).. hence I need new plotting symbols
> please.. I would appreciate it if somebody could point me to a
> database..

Exponentially! Yikes!

Here is a pretty good database of interesting symbols:

http://altreligion.about.com/library/glossary/bloccultsymbol s.htm

And you could look at the SYMBOL method in MPI_PlotConfig__Define
for 18 more. I guess I could think of a few more: three-quarter
filled/unfilled circle, half filled/unfilled circle, etc.
I'd be hard pressed to think of 30, let alone 1000's. But
maybe that will get you started. I think users can only
keep about six in mind at any one time anyway, so maybe
you can finesse it. :-)

http://www.dfanning.com/programs/mpi_plotconfig__define.pro

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui.
(Opata Indian saying, meaning "Perhaps thou speakest truth.")

Subject: Re: need more plotting symbols please
Posted by Matthias Cuntz on Fri, 01 Sep 2006 04:35:49 GMT
View Forum Message <> Reply to Message

Dear Dilkushi,

use the routine below.

Cheers,
Matthias.

```
;+
; NAME:
;    MC_SYMBOL
;
; PURPOSE:
;     Create user defined plotting symbol.
;
; CATEGORY:
;    Graphics.
;
; CALLING SEQUENCE:
;       mc_symbol, sym=sym, fill=fill, thick=thick, help=help
;
; INPUTS:
;    sym:    Integer specifying the symbol type (see below).
;    *** MC_SYMBOL TYPE ***
;    1         ->    square
;    2         ->    circle
;    3         ->    triangle
;    4         ->    triangle (inverted)
;    5         ->    diamond
;    6         ->    star 1
;    7         ->    star 2
;    8         ->    hourglass
;    9         ->    bowtie
;    10         ->     hourglass tilted left
;    11         ->     hourglass tilted right
;    12         ->     rigth half circel
;    13         ->     left half circel
;    14         ->     upper half circel
;    15         ->     lower half circel
;    16         ->     standing bar
;    17         ->     laying bar
;    18         ->     hat up
;    19         ->     hat down
;    20         ->     hat right
;    21         ->     hat left
;    22         ->     big cross
;    23         ->     circle w/ plus
;    24         ->     circle w/ ex
;    25         ->     partial arrow (right)
;    26         ->     partial arrow (left)
;    27         ->     full arrow
;    28         ->     vertical line
;    29         ->     horizontal line
;    30         ->     x
;    31         ->     +
```

```
;
; OPTIONAL INPUT PARAMETERS:
;    fill:    If specified then symbol is filled.
;    thick:    Specifies the line thickness of symbol.  Default:  thick=1.0
;
; OUTPUTS:
;    Defines IDL USERSYMBOL PSYM=8.
;
; COMMON BLOCKS:
;    None.
;
; SIDE EFFECTS:
;    None.
;
; RESTRICTIONS:
;    None.
;
; PROCEDURE:
;       MC_SYMBOL,sym=5           <- open diamond
;       PLOT,x,y,PSYM=8,COLOR=col
;       MC_SYMBOL,sym=2,fill=1           <- filled circle
;       PLOT,x,y,PSYM=8,SYMSIZE=.8,COLOR=col
;
; MODIFICATION HISTORY:
;    Written, KAM, Apr. 1993.
;       Modified, MC, Mar. 2002 - include sym=0 as no symbol; sym=0 is a
tiny little square
;                 MC, July 2002 - 11 new symbols which can be filled
;                                 - change order: all fillable symbols first
;                                 - fallback is sym=0
;-
PRO MC_SYMBOL, sym=sym, fill=fill, thick=thick, help=help
  COMPILE_OPT idl2
  ON_ERROR,2
  ;
  if keyword_set(help) then begin
     print,"mc_symbol, sym=sym, /fill, thick=thick, /help"
     print,"Symbol types: "
     print,"  sym=1        ->    square"
     print,"  sym=2        ->    circle"
     print,"  sym=3        ->    triangle"
     print,"  sym=4        ->    triangle (inverted)"
     print,"  sym=5        ->    diamond"
     print,"  sym=6        ->    star 1"
     print,"  sym=7        ->    star 2"
     print,"  sym=8        ->    hourglass"
     print,"  sym=9        ->    bowtie"
     print,"  sym=10       ->    hourglass tilted left"
```

```
      print,"  sym=11       ->   hourglass tilted right"
      print,"  sym=12       ->   rigth half circel"
      print,"  sym=13       ->   left half circel"
      print,"  sym=14       ->   upper half circel"
      print,"  sym=15       ->   lower half circel"
      print,"  sym=16       ->   standing bar"
      print,"  sym=17       ->   laying bar"
      print,"  sym=18       ->   hat up"
      print,"  sym=19       ->   hat down"
      print,"  sym=20       ->   hat right"
      print,"  sym=21       ->   hat left"
      print,"  sym=22       ->   big cross"
      print,"  sym=23       ->   circle w/ plus"
      print,"  sym=24       ->   circle w/ ex"
      print,"  sym=25       ->   partial arrow (right)"
      print,"  sym=26       ->   partial arrow (left)"
      print,"  sym=27       ->   full arrow"
      print,"  sym=28       ->   vertical line"
      print,"  sym=29       ->   horizontal line"
      print,"  sym=30       ->   x"
      print,"  sym=31       ->   +"
      print,"include fill=1 for filled symbol, thick=thick for line
thickness."
      return
   endif
   ;
   ; For sym=0
   if n_elements(sym) eq 0 then sym=0 else sym=fix(sym)
   if not keyword_set(fill) then fill=0
   if n_elements(thick) eq 0 then thick=1.
   ;
   case sym of
      ;
      ; tiny little square
      ;
      0: begin
         x=[-1,-1, 1, 1,-1]
         y=[-1, 1, 1,-1,-1]
         x=x/10000.
         y=y/10000.
      end
      ;
      ; square
      ;
      1: begin
         x=[-1,-1, 1, 1,-1]
         y=[-1, 1, 1,-1,-1]
      end
```

```
;
; circle
;
2: begin
   x=[1,.866, .707, .500, 0,-.500,-.707,-.866,-1,$
     -.866,-.707,-.500, 0, .500, .707, .866, 1]
   y=[0,.500, .707, .866, 1, .866, .707, .500, 0,$
     -.500,-.707,-.866,-1,-.866,-.707,-.500, 0]
end
;
; triangle
;
3: begin
   x=[-0.924, 0.000, 0.924,-0.924]
   y=[-0.600, 1.000,-0.600,-0.600]
end
                        ;
; inverted triangle
;
4: begin
   x=[-0.924, 0.000, 0.924,-0.924]
   y=[ 0.600,-1.000, 0.600, 0.600]
end
;
; diamond
;
5: begin
   x=[ 0,-1, 0, 1, 0]
   y=[-1, 0, 1, 0,-1]
end
;
; star 2
;
6: begin
   x=[-1,-.2, 0,.2,1, .2, 0,-.2,-1]
   y=[ 0, .2, 1,.2,0,-.2,-1,-.2, 0]
end
;
; star 1
;
7: begin
   x=[ 0, .4,  1, .7,  1, .4, 0, -.4, -1, -.7,  -1, -.4, 0]
   y=[-1,-.4,-.4,  0, .4, .4, 1,  .4, .4,   0, -.4, -.4,-1]
end
;
; hourglass
;
8: begin
```

```
   x=[-1, 1,-1,1,-1]
   y=[-1,-1, 1,1,-1]
end
;
; hourglass (on side)
;
9: begin
   x=[-1,-1, 1,1,-1]
   y=[-1, 1,-1,1,-1]
end
;
; hourglass tilted left
;
10: begin
   x=[-1, 0, 0,-1, 1, 0, 0, 1,-1]
   y=[ 0, 1, 0, 0, 0,-1, 0, 0, 0]
end
;
; hourglass tilted right
;
11: begin
   x=[-1, 1, 0, 0,-1]
   y=[ 0, 0, 1,-1, 0]
end
;
; rigth half circel
;
12: begin
   x=[1,.866, .707, .500, 0,$
     0, .500, .707, .866, 1]
   y=[0,.500, .707, .866, 1,$
     -1,-.866,-.707,-.500, 0]
end
;
; left half circel
;
13: begin
   x=[0,-.500,-.707,-.866,-1,$
     -.866,-.707,-.500, 0]
   y=[1, .866, .707, .500, 0,$
     -.500,-.707,-.866,-1]
end
;
; upper half circel
;
14: begin
   x=[1,.866, .707, .500, 0,-.500,-.707,-.866,-1,1]
   y=[0,.500, .707, .866, 1, .866, .707, .500, 0,0]
```

```
end
;
; lower half circel
;
15: begin
   x=[1,.866, .707, .500, 0,-.500,-.707,-.866,-1,1]
   y=[0,-.500,-.707,-.866,-1,-.866,-.707,-.500,0,0]
end
;
; standing bar
;
16: begin
   x=[-0.5,-0.5, 0.5, 0.5,-0.5]
   y=[-1, 1, 1,-1,-1]
end
;
; laying bar
;
17: begin
   x=[-1,-1, 1, 1,-1]
   y=[-0.5, 0.5, 0.5,-0.5,-0.5]
end
;
; hat up
;
18: begin
   x=[-1, -0.5, -0.5, 0.5, 0.5, 1, -1]
   y=[-0.7, -0.7, 0.7, 0.7, -0.7, -0.7, -0.7]
end
;
; hat down
;
19: begin
   x=[-1, -0.5, -0.5, 0.5, 0.5, 1, -1]
   y=[0.7, 0.7, -0.7, -0.7, 0.7, 0.7, 0.7]
end
;
; hat right
;
20: begin
   x=[-0.7, -0.7, 0.7, 0.7, -0.7, -0.7, -0.7]
   y=[-1, -0.5, -0.5, 0.5, 0.5, 1, -1]
end
;
; hat left
;
21: begin
   x=[0.7, 0.7, -0.7, -0.7, 0.7, 0.7, 0.7]
```

```
    y=[-1, -0.5, -0.5, 0.5, 0.5, 1, -1]
end
;
; big cross
;
22: begin
    x=[1, 0.3, 0.3, -0.3, -0.3, -1, -1, -0.3, -0.3, 0.3, 0.3, 1, 1]
    y=[0.3, 0.3, 1, 1, 0.3, 0.3, -0.3, -0.3, -1, -1, -0.3, -0.3, 0.3]
end
;
; circle and plus
;
23: begin
    x=[1,.866, .707, .500, 0,-.500,-.707,-.866,-1,$
       -.866,-.707,-.500, 0, .500, .707, .866, 1,$
       -1,   0,    0,    0]
    y=[0,.500, .707, .866, 1, .866, .707, .500, 0,$
       -.500,-.707,-.866,-1,-.866,-.707,-.500, 0,$
       0,   0,    1,   -1]
end
;
; circle and ex
;
24: begin
    x=[1,.866, .707, .500, 0,-.500,-.707,-.866,-1,$
       -.866,-.707,-.500, 0, .500, .707, .866, 1,$
       .866,.707,-.707,    0, .707,-.707]
    y=[0,.500, .707, .866, 1, .866, .707, .500, 0,$
       -.500,-.707,-.866,-1,-.866,-.707,-.500, 0,$
       .500,.707,-.707,    0,-.707, .707]
end
;
; right arrow
;
25: begin
    x=[ 0.000,0.000,0.342,0.000]
    y=[-1.000,1.000,0.000,0.000]
end
;
; left arrow
;
26: begin
    x=[ 0.000,0.000,-0.342,0.000]
    y=[-1.000,1.000, 0.000,0.000]
end
;
; full arrow
;
```

```
      27: begin
         x=[ 0.000,0.000, 0.342,-0.342,0.000]
         y=[-1.000,1.000, 0.000,0.000,1.000]
      end
      ;
      ; vertical line
      ;
      28: begin
         x=[ 0.000,0.000]
         y=[-1.000,1.000]
      end
      ;
      ; horizontal line
      ;
      29: begin
         x=[-1.000,1.000]
         y=[ 0.000,0.000]
      end
      ;
      ; x
      ;
      30: begin
         x=[-1, 0, 1, 0, 1, 0, -1, 0, -1]
         y=[1, 0, 1, 0, -1, 0, -1, 0, 1]
      end
      ;
      ; +
      ;
      31: begin
         x=[0,0,0,0,-1,0,1]
         y=[1,0,-1,0,0,0,0]
      end
      ; fall back is like sym=0
      else: begin
         x=[-1,-1, 1, 1,-1]
         y=[-1, 1, 1,-1,-1]
         x=x/10000.
         y=y/10000.
      end
   endcase
   ;
   usersym, x, y, fill=fill, thick=thick
   ;
   return
   ;
END
;-
```

Subject: Re: need more plotting symbols please
Posted by larkn10 on Fri, 01 Sep 2006 19:06:37 GMT
View Forum Message <> Reply to Message

When I am in a similar situation my plots get pretty cluttered. But anyway,
if thirty-some symbols are not good enough, you can have a parallel set of
colors and wrap through those as well.

e.g. 10 colors and 31 symbols gives you 310 options. (Is my math right?)

Subject: Re: need more plotting symbols please
Posted by Dilkushi@gmail.com on Fri, 01 Sep 2006 20:46:34 GMT
View Forum Message <> Reply to Message

Thanks David
it works just great,,,
d
David Fanning wrote:

> Dilkushi@gmail.com writes:
>
>> In my assignment the number of variables I am requested to plot are
>> increasing (exponentially !!).. hence I need new plotting symbols
>> please.. I would appreciate it if somebody could point me to a
>> database..
>
> Exponentially! Yikes!
>
> Here is a pretty good database of interesting symbols:
>
> http://altreligion.about.com/library/glossary/bloccultsymbol s.htm
>
> And you could look at the SYMBOL method in MPI_PlotConfig__Define
> for 18 more. I guess I could think of a few more: three-quarter
> filled/unfilled circle, half filled/unfilled circle, etc.
> I'd be hard pressed to think of 30, let alone 1000's. But
> maybe that will get you started. I think users can only
> keep about six in mind at any one time anyway, so maybe
> you can finesse it. :-)
>
> http://www.dfanning.com/programs/mpi_plotconfig__define.pro
>
> Cheers,
>

> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.dfanning.com/
> Sepore ma de ni thui.
> (Opata Indian saying, meaning "Perhaps thou speakest truth.")

Subject: Re: need more plotting symbols please
Posted by David Fanning on Sat, 02 Sep 2006 18:18:31 GMT
View Forum Message <> Reply to Message

larkn10 writes:

> When I am in a similar situation my plots get pretty cluttered. But
> anyway,
> if thirty-some symbols are not good enough, you can have a parallel set
> of
> colors and wrap through those as well.
>
> e.g. 10 colors and 31 symbols gives you 310 options. (Is my math
> right?)

I wrote a small function that combines some of the things
I found in MC_SYMBOL with some ideas I had developed in
MPI_PLOTCONFIG. There are 44 symbols in this symbol catalog
function. The return value of the function is used as
input to the PSYM keyword in graphics commands. For example,
to use an hourglass symbol (symbol number 19), you would
type this:

```
Plot, findgen(11), Symsize=2.0, PSym=SYMCAT(19)
```

You can find the program here:

http://www.dfanning.com/programs/symcat.pro

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui.
(Opata Indian saying, meaning "Perhaps thou speakest truth.")

Subject: Re: need more plotting symbols please
Posted by Dilkushi@gmail.com on Tue, 05 Sep 2006 16:04:18 GMT
View Forum Message <> Reply to Message

Thank you Dr. Fanning
this is excellent
thanks
d
David Fanning wrote:

> larkn10 writes:
>
>> When I am in a similar situation my plots get pretty cluttered. But
>> anyway,
>> if thirty-some symbols are not good enough, you can have a parallel set
>> of
>> colors and wrap through those as well.
>>
>> e.g. 10 colors and 31 symbols gives you 310 options. (Is my math
>> right?)
>
> I wrote a small function that combines some of the things
> I found in MC_SYMBOL with some ideas I had developed in
> MPI_PLOTCONFIG. There are 44 symbols in this symbol catalog
> function. The return value of the function is used as
> input to the PSYM keyword in graphics commands. For example,
> to use an hourglass symbol (symbol number 19), you would
> type this:
>
> Plot, findgen(11), Symsize=2.0, PSym=SYMCAT(19)
>
> You can find the program here:
>
> http://www.dfanning.com/programs/symcat.pro
>
> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.dfanning.com/
> Sepore ma de ni thui.
> (Opata Indian saying, meaning "Perhaps thou speakest truth.")

Subject: Re: need more plotting symbols please
Posted by don.woodraska on Mon, 02 Oct 2006 16:17:54 GMT

David,

I modified your code, symcat.pro, to support negative symbols. For those who don't know, negative symbols produce lines that connect the points in a plot. Thank you David for writing clean code that allowed me to only change 3 lines.

Don Woodraska

Save this file with the same name as David's symcat.pro

```
;+
; NAME:
;       SYMCAT
;
; PURPOSE:
;
;       This function provides a symbol catalog for specifying a number
of plotting symbols.
;
; AUTHOR:
;
;       FANNING SOFTWARE CONSULTING
;       David Fanning, Ph.D.
;       1645 Sheely Drive
;       Fort Collins, CO 80526 USA
;       Phone: 970-221-0438
;       E-mail: davidf@dfanning.com
;       Coyote's Guide to IDL Programming: http://www.dfanning.com
;
; CATEGORY:
;
;       Graphics
;
; CALLING SEQUENCE:
;
;       Plot, findgen(11), PSYM=SYMCAT(theSymbol)
;
; INPUTS:
;
;       theSymbol:    The number of the symbol you wish to use.
Possible values are:
;
;       0  : No symbol.
;       1  : Plus sign.
;       2  : Asterisk.
;       3  : Dot (period).
```

```
;       4  : Open diamond.
;       5  : Open upward triangle.
;       6  : Open square.
;       7  : X.
;       8  : Open circle.
;       9  : Open downward triangle.
;      10  : Open rightfacing triangle.
;      11  : Open leftfacing triangle.
;      12  : Filled diamond.
;      13  : Filled square.
;      14  : Filled circle.
;      15  : Filled upward triangle.
;      16  : Filled downward triangle.
;      17  : Filled rightfacing triangle.
;      18  : Filled leftfacing triangle.
;      19  : Hourglass.
;      20  : Filled Hourglass.
;      21  : Bowtie.
;      22  : Filled bowtie.
;      23  : Standing Bar.
;      24  : Filled Standing Bar.
;      25  : Laying Bar.
;      26  : Filled Laying Bar.
;      27  : Hat up.
;      28  : Hat down.
;      29  : Hat right.
;      30  : Hat down.
;      31  : Big cross.
;      32  : Filled big cross.
;      33  : Circle with plus.
;      34  : Circle with X.
;      35  : Upper half circle.
;      36  : Filled upper half circle.
;      37  : Lower half circle.
;      38  : Filled lower half circle.
;      39  : Left half circle.
;      40  : Filled left half circle.
;      41  : Right half circle.
;      42  : Filled right half circle.
;      43  : Star.
;      44  : Filled star.
;
; RETURN VALUE:
;
;       The return value is whatever is appropriate for passing along
;       to the PSYM keyword of (for example) a PLOT or OPLOT command.
;       For the vast majority of the symbols, the return value is 8.
;
```

```
; KEYWORDS:
;
;       None.
;
; MODIFICATION HISTORY:
;
;       Written by David W. Fanning, 2 September 2006. Loosely based on
the
;         program MC_SYMBOL introduced on the IDL newsgroup 1
September 2006,
;         and MPI_PLOTCONFIG__DEFINE from the Coyote Library.
;
;       Added support for negative arguments (connected symbols) Don
Woodraska
;         September 2, 2006
;-
;
 ;############################################################################
;
; LICENSE
;
; This software is OSI Certified Open Source Software.
; OSI Certified is a certification mark of the Open Source Initiative.
;
; Copyright © 2006 Fanning Software Consulting.
;
; This software is provided "as-is", without any express or
; implied warranty. In no event will the authors be held liable
; for any damages arising from the use of this software.
;
; Permission is granted to anyone to use this software for any
; purpose, including commercial applications, and to alter it and
; redistribute it freely, subject to the following restrictions:
;
; 1. The origin of this software must not be misrepresented; you must
;    not claim you wrote the original software. If you use this
software
;    in a product, an acknowledgment in the product documentation
;    would be appreciated, but is not required.
;
; 2. Altered source versions must be plainly marked as such, and must
;    not be misrepresented as being the original software.
;
; 3. This notice may not be removed or altered from any source
distribution.
;
; For more information on Open Source Software, visit the Open Source
; web site: http://www.opensource.org.
```

```
;
 ;############################################################## ################

FUNCTION SymCat, theSymbol

   On_Error, 2

   ; Error checking.
   IF N_Elements(theSymbol) EQ 0 THEN RETURN, 0
   IF (abs(theSymbol) GT 44) THEN Message, 'Symbol number out of
defined range.'
   theSymbol = Fix(theSymbol)

   ; Define helper variables for creating circles.
   phi = Findgen(36) * (!PI * 2 / 36.)
   phi = [ phi, phi(0) ]

   ; Use user defined symbol by default.
   result = 8

   CASE abs(theSymbol) OF

      0  : result = 0
   ; No symbol.
      1  : result = 1
   ; Plus sign.
      2  : result = 2
   ; Asterisk.
      3  : result = 3
   ; Dot (period).
      4  : UserSym, [ 0, 1, 0, -1, 0 ], [ 1, 0, -1, 0, 1 ]
   ; Open diamond.
      5  : UserSym, [ -1, 0, 1, -1 ], [ -1, 1, -1, -1 ]
   ; Open upward triangle.
      6  : UserSym, [ -1, 1, 1, -1, -1 ], [ 1, 1, -1, -1, 1 ]
   ; Open square.
      7  : result = 7
   ; X.
      8  : UserSym, cos(phi), sin(phi)
   ; Open circle.
      9  : UserSym, [ -1, 0, 1, -1 ], [  1, -1, 1, 1 ]
   ; Open downward triangle.
     10  : UserSym, [ -1, 1, -1, -1 ], [1, 0, -1, 1 ]
   ; Open rightfacing triangle.
     11  : UserSym, [ 1, -1, 1, 1 ], [1, 0, -1, 1 ]
   ; Open leftfacing triangle.
     12  : UserSym, [ 0, 1, 0, -1, 0 ], [ 1, 0, -1, 0, 1 ], /Fill
   ; Filled diamond.
```

```
    13  : UserSym, [ -1, 1, 1, -1, -1 ], [ 1, 1, -1, -1, 1 ], /Fill
  ; Filled square.
    14  : UserSym, Cos(phi), Sin(phi), /Fill
  ; Filled circle.
    15  : UserSym, [ -1, 0, 1, -1 ], [ -1, 1, -1, -1 ], /Fill
  ; Filled upward triangle.
    16  : UserSym, [ -1, 0, 1, -1 ], [  1, -1, 1, 1 ], /Fill
  ; Filled downward triangle.
    17  : UserSym, [ -1, 1, -1, -1 ], [1, 0, -1, 1 ], /Fill
  ; Filled rightfacing triangle.
    18  : UserSym, [ 1, -1, 1, 1 ], [1, 0, -1, 1 ], /Fill
  ; Filled leftfacing triangle.
    19  : UserSym, [-1, 1,-1,1,-1], [-1,-1, 1,1,-1]
  ; Hourglass.
    20  : UserSym, [-1, 1,-1,1,-1], [-1,-1, 1,1,-1], /Fill
  ; Filled Hourglass.
    21  : UserSym, [-1,-1, 1,1,-1], [-1, 1,-1,1,-1]
  ; Bowtie.
    22  : UserSym, [-1,-1, 1,1,-1], [-1, 1,-1,1,-1], /Fill
  ; Filled bowtie.
    23  : UserSym, [-0.5,-0.5, 0.5, 0.5,-0.5], [-1, 1, 1,-1,-1]
  ; Standing Bar.
    24  : UserSym, [-0.5,-0.5, 0.5, 0.5,-0.5], [-1, 1, 1,-1,-1],
/Fill   ; Filled Standing Bar.
    25  : UserSym, [-1,-1, 1, 1,-1], [-0.5, 0.5, 0.5,-0.5,-0.5]
  ; Laying Bar.
    26  : UserSym, [-1,-1, 1, 1,-1], [-0.5, 0.5, 0.5,-0.5,-0.5],
/Fill   ; Filled Laying Bar.
    27  : UserSym, [-1, -0.5, -0.5, 0.5, 0.5, 1, -1], [-0.7, -0.7,
0.7, 0.7, -0.7, -0.7, -0.7] ; Hat up.
    28  : UserSym, [-1, -0.5, -0.5, 0.5, 0.5, 1, -1], [0.7, 0.7,
-0.7, -0.7, 0.7, 0.7, 0.7]    ; Hat down.
    29  : UserSym, [-0.7, -0.7, 0.7, 0.7, -0.7, -0.7, -0.7], [-1,
-0.5, -0.5, 0.5, 0.5, 1, -1] ; Hat right.
    30  : UserSym, [0.7, 0.7, -0.7, -0.7, 0.7, 0.7, 0.7], [-1, -0.5,
-0.5, 0.5, 0.5, 1, -1]    ; Hat down.
    31  : UserSym, [1, 0.3, 0.3, -0.3, -0.3, -1, -1, -0.3, -0.3, 0.3,
0.3, 1, 1], $
                [0.3, 0.3, 1, 1, 0.3, 0.3, -0.3, -0.3, -1, -1,
-0.3, -0.3, 0.3]            ; Big cross.
    32  : UserSym, [1, 0.3, 0.3, -0.3, -0.3, -1, -1, -0.3, -0.3, 0.3,
0.3, 1, 1], $
                [0.3, 0.3, 1, 1, 0.3, 0.3, -0.3, -0.3, -1, -1,
-0.3, -0.3, 0.3], /Fill      ; Filled big cross.
    33  : UserSym, [1,.866, .707, .500, 0,-.500,-.707,-.866,-1, $
                 -.866,-.707,-.500, 0, .500, .707, .866, 1, -1, 0,
0, 0], $
                [0,.500, .707, .866, 1, .866, .707, .500, 0, $
```

```
                -.500,-.707,-.866,-1,-.866,-.707,-.500, 0, 0, 0,
1, -1]                  ; Circle with plus.
     34  : UserSym, [1,.866, .707, .500, 0,-.500,-.707,-.866,-1, $
                -.866,-.707,-.500, 0, .500, .707, .866, 1, $
                .866,.707,-.707,    0, .707,-.707], $
               [0,.500, .707, .866, 1, .866, .707, .500, 0, $
                -.500,-.707,-.866,-1,-.866,-.707,-.500, 0, $
                .500,.707,-.707,    0,-.707, .707]
                  ; Circle with X.
     35  : UserSym, [Cos(phi[0:18]), Cos(phi[0])], [Sin(phi[0:18]),
Sin(phi[0])]-0.5           ; Upper half circle.
     36  : UserSym, [Cos(phi[0:18]), Cos(phi[0])], [Sin(phi[0:18]),
Sin(phi[0])]-0.5, /Fill    ; Filled upper half circle.
     37  : UserSym, [Cos(phi[18:36]), Cos(phi[18])], [Sin(phi[18:36]),
Sin(phi[18])]+0.5       ; Lower half circle.
     38  : UserSym, [Cos(phi[18:36]), Cos(phi[18])], [Sin(phi[18:36]),
Sin(phi[18])]+0.5, /Fill ; Filled lower half circle.
     39  : UserSym, [Cos(phi[9:27]), Cos(phi[9])]+0.5,
[Sin(phi[9:27]), Sin(phi[9])]          ; Left half circle.
     40  : UserSym, [Cos(phi[9:27]), Cos(phi[9])]+0.5,
[Sin(phi[9:27]), Sin(phi[9])], /Fill    ; Filled left half circle.
     41  : UserSym, [Cos(phi[27:36]), Cos(phi[0:9]),
Cos(phi[27])]-0.5, $
               [Sin(phi[27:36]), Sin(phi[0:9]), Sin(phi[27])]
                  ; Right half circle.
     42  : UserSym, [Cos(phi[27:36]), Cos(phi[0:9]),
Cos(phi[27])]-0.5, $
               [Sin(phi[27:36]), Sin(phi[0:9]), Sin(phi[27])],
/Fill                   ; Filled right half circle.
     43  : UserSym, [-1,-.33, 0,.33,1, .33, 0,-.33,-1], $
               [ 0, .33, 1,.33,0,-.33,-1,-.33, 0]
                  ; Star.
     44  : UserSym, [-1,-.33, 0,.33,1, .33, 0,-.33,-1], $
               [ 0, .33, 1,.33,0,-.33,-1,-.33, 0], /Fill
                  ; Filled star.
   ENDCASE

   if theSymbol gt 0 then theSign = 1 else theSign = -1
   RETURN, result * theSign
END
 ;------------------------------------------------------------ ------------------------------------------------------------ ------
```