
Subject: Polar plots

Posted by [jchapman](#) on Tue, 20 Aug 1996 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

Does anybody have code written to plot polar data (i.e. to fill sectors of radius r to $r + dr$ and angle θ to $\theta + d\theta$) using IDL? I can't seem to find anything like this built into the language. I've tried using the MAP_SET routines with some success, but it seems like there should be a better way to do this. Any suggestions?

Subject: Re: Polar Plots

Posted by [Dr. Otto Jusko](#) on Thu, 26 Apr 2001 09:25:49 GMT

[View Forum Message](#) <> [Reply to Message](#)

I have written a package for analysing data of form profiles i.e. straightness and roundness profiles. For the roundness case extended polar plotting is needed. I don't think that it can be used without modification. But the principle might be of interest.

Basically it works like this:

A plot center is defined and the data is converted to cartesian coordinates. After that a command like the following is executed:

```
winkel = array of angular steps
rawdat = polar data
auswfakt = scaling factor
...
rect_dat = cv_coord (from_pol=transpose([[winkel],[rawdat* auswfakt,
rawdat[0]* auswfakt]]),/to_rect)
radius = fltarr(N_elements(winkel)) + plotradius
radius = cv_coord (from_pol=transpose([[winkel],[radius]]),/to_rect)
rect_dat = rect_dat + radius
...
rect_dat[0,*] = rect_dat[0,*] + plot_center[0]
rect_dat[1,*] = rect_dat[1,*] + plot_center[1]
..
plots, noclip=0, rect_dat,/device
```

For circular axes the following is performed: A max. and min. radius is defined. Coordinates of the vectors for the axes are calculated.

```
zwinkel = array of angular steps
LinCnt = Counter for several plot circles
add25 = a small value
```

```
maxradius = fltarr(N_elements(zwinkel)) + maxrad[LinCnt]-add25
maxradius = cv_coord (from_pol=transpose([[zwinkel],[maxradius]]),/to_rect)
maxradius[0,*] = maxradius[0,*] + plot_mitte[0]
maxradius[1,*] = maxradius[1,*] + plot_mitte[1]
minradius = fltarr(N_elements(zwinkel)) + minrad[LinCnt]+add25
minradius = cv_coord (from_pol=transpose([[zwinkel],[minradius]]),/to_rect)
minradius[0,*] = minradius[0,*] + plot_mitte[0]
minradius[1,*] = minradius[1,*] + plot_mitte[1]
```

After that everything is plotted in a single command:

```
plots, noclip=0, rect_dat,/device
```

Otto

Chris Bull schrieb:

```
> Hi all,
>
> I was just wondering if anyone had extended (and is willing to share) the
> IDL basic polar plot routine to include useful things like circular axes
> etc (before I slave over doing it myself! :*)
>
> Regards
>
> Chris Bull
```

File Attachments

1) [Otto.Jusko.vcf](#), downloaded 149 times

Subject: Re: Polar Plots

Posted by [davidf](#) on Thu, 26 Apr 2001 09:41:36 GMT

[View Forum Message](#) <> [Reply to Message](#)

Chris Bull (cjbull@another.com) writes:

```
> I was just wondering if anyone had extended (and is willing to share) the
> IDL basic polar plot routine to include useful things like circular axes
> etc (before I slave over doing it myself! :*)
```

I have a simple of example of this on my web page:

http://www.dfanning.com/tips/polar_plot.html

It is really not too hard. I don't think I would
"slave" over it too long. :-)

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Polar Plots
Posted by [Chris Bull](#) on Thu, 26 Apr 2001 21:43:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

Cheers!

I figured it wasnt going to be `_that_` hard! :) but if someone's
already cleared a path why hack thru jungle!

regards

Chris

"David Fanning" <davidf@dfanning.com> wrote in message
news:MPG.15519b3e37c0166d989df0@news.frii.com...
> Chris Bull (cjbull@another.com) writes:
>
>> I was just wondering if anyone had extended (and is willing to share)
the
>> IDL basic polar plot routine to include useful things like circular axes
>> etc (before I slave over doing it myself! :*)
>
> I have a simple of example of this on my web page:
>
> http://www.dfanning.com/tips/polar_plot.html
>
> It is really not too hard. I don't think I would
> "slave" over it too long. :-)
>
> Cheers,

>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting
> Phone: 970-221-0438 E-Mail: davidf@dfanning.com
> Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
> Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Polar Plots
Posted by [Paul Krummel](#) on Fri, 27 Apr 2001 02:40:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Chris,

I have written some code to plot what I was calling a wind rose (It's not a real wind rose, just a routine to plot a value against a direction, which in this case is wind direction). Anyway, you might want to have a look at the code and modify it for your needs. Note that 0 degrees in this plot is at "north" or top y-axis whereas in true polar coords 0 degrees is right hand x-axis. The code does a lot of other things that you are probably not interested in eg can plot coastlines behind the polar plot. Find the code plot_wind_rose.pro attached. Hope this helps!

Cheers, Paul

"Chris Bull" <cjbull@another.com> wrote in message
news:sCmF6.8777\$Ln6.1258097@news2-win.server.ntlworld.com...

> Hi all,
>
>
> I was just wondering if anyone had extended (and is willing to share) the
> IDL basic polar plot routine to include useful things like circular axes
> etc (before I slave over doing it myself! :*)
>
> Regards
>
>
> Chris Bull
>
>

FUNCTION CIRCLE, xcenter, ycenter, radius, nPts=nPts

```
IF N_Elements(nPts) EQ 0 THEN nPts = 100
points = (2 * !PI / (nPts-1)) * FIndGen(nPts)
x = xcenter + radius * Cos(points)
y = ycenter + radius * Sin(points)
RETURN, Transpose([[x],[y]])
```

```
END
```

```
;+
```

```
; NAME:
```

```
; PLOT_WIND_ROSE
```

```
;
```

```
; PURPOSE:
```

```
; This procedure will plot a given variable against  
; wind direction in a polar plot. This type of plot  
; is commonly known as a wind rose in meteorology.  
; The routine has several options including the option  
; of plotting the rose centred over a particular map  
; location. Other options include placing baseline  
; indicators, any number of circles and cross hairs  
; onto the plot.
```

```
;
```

```
; CATEGORY:
```

```
; Plotting/meteorology.
```

```
;
```

```
; CALLING SEQUENCE:
```

```
; PLOT_WIND_ROSE, Var, Wind_Dirn
```

```
;
```

```
; INPUTS:
```

```
; Var: This is the variable to plot on the wind rose.  
; This is an array of the same length as Wind_Dirn  
; corresponding to each element in Wind_Dirn.  
; Wind_Dirn: This is the wind direction in degrees.
```

```
;
```

```
; KEYWORD PARAMETERS:
```

```
; TITLE: A string that contains a title for the plot.  
; COLOR: This controls the colour of the wind rose line.  
; Default color is 0.  
; THICK: This controls the thickness of the wind rose  
; line. Default thickness is 4.  
; LINestyle: This controls the linestyle of the wind rose  
; line. Default linestyle is 0.
```

```
;
```

```
; MAXV: Set this keyword to a scalar of the maximum  
; value of the plot axis. Positive scalar only.
```

```
;
```

```
; VAR2: Set this to an array to plot a second wind rose.  
; This must be an array of the same length as WIND_DIRN2  
; corresponding to each element in WIND_DIRN2.
```

```

; WIND_DIRN2: This is the second wind direction in degrees.
; V2_COLOR: This controls the colour of the 2nd wind rose line.
; Default color is 0.
; V2_THICK: This controls the thickness of the 2nd wind rose
; line. Default thickness is 4.
; V2_LINestyle: This controls the linestyle of the 2nd wind rose
; line. Default linestyle is 1.
;
; YTITLE: A string that contains a title for the y-axis.
; YTHICK: This controls the thickness of the y-axis.
; Default thickness is 2.
; YMINOR: This controls the number of minor tickmarks.
; Default is 5.
;
; CROSS_HAIRS: Set this keyword to place tick marks on
; the horizontal and vertical radial lines through
; the centre of the plot.
; E_CROSS: Set this keyword to a structure that will
; contain any keywords used in the AXIS procedure.
; The AXIS procedure is used to draw the cross hairs.
;
; N_RAD_LINES: Set this keyword to the number of radial
; lines ("spokes") that will be plotted and labelled.
; If not set at all then the default is to plot 12 radial
; lines (ie every 30 degrees). Set N_RAD_LINES to zero to
; plot NO radial lines. If set to zero and CROSS_HAIRS is
; selected then the cross hair angles are labelled (ie 0,
; 90, 180 and 270 degrees).
; E_RAD_LINES: Set this keyword to a structure that will
; contain any keywords (linestyle, color, thickness etc)
; used in the OPLOT procedure for plotting the radial
; lines. Default is same as for OPLOT and THICK=1.0.
; NOTE: If CROSS_HAIRS is selected and the angles
; for the radial lines coincide with the cross hairs
; angles then the radial lines are plotted OVER the
; cross hair lines!
;
; BASELINE: Set this keyword if baseline indicator lines
; are required. The default is plot the baseline
; indicators for Cape Grim, Tasmania ie lines at 190
; and 280 degrees. To change this, set this keyword to
; an array of the angles required ie BASELINE=[150.,290.]
; This array can contain any number of angles as long
; as there is more than 1 angle.
; E_BASE: Set this keyword to a structure that will contain
; any keywords used in the OPLOT procedure ie linestyle,
; color, thickness etc. Defaults are the same as used
; in OPLOT.

```

```

;
; MAP: Set this keyword to place a map on the wind rose.
; The default is to plot a high resolution map with the
; COASTS keyword set centred on Cape Grim, Tasmania
; (~-40.68,~144.68). To centre the map on a different
; location set the keyword to a two element array
; containing the latitude and longitude of the desired
; point ie MAP=[-38.03,145.10]. The map is setup with
; MAP_SET and drawn with MAP_CONTINENTS.
; E_MAP: Set this keyword to a structure that will contain
; any keywords used in the MAP_SET procedure. Use this
; to set the map scale or projection. The map scale
; controls what the map region is. The default is 5.e6
; so a map of scale 1:5e6 is plotted centred on the given
; lat and lon. DO NOT use this to set actual plotting
; features ie color, thickness, fill etc Use the E_CONT
; keyword described below.
; E_CONT: Set this keyword to a structure that will contain
; any keywords used in the MAP_CONTINENTS procedure. Use
; this to set plotting variables of the map such as line
; thickness, color, turn high res/coasts off, etc.
; GRID: Set this keyword to plot a grid on the map using
; MAP_GRID.
; E_GRID: Set this keyword to a structure that will contain
; any keywords used in the MAP_GRID procedure. Use this
; keyword to set the linestyle, thickness, color, labels
; etc for the grid. Refer to MAP_GRID.
;
; N_CIRC: Set this keyword to an array containing values
; between zero and one indicating where to plot the
; circles on the wind rose. The default if this is not
; set is to plot two circles, one at 0.5 of the maximum
; value, the other at 1. If you wanted to plot 4 circles
; set this keyword to where the circles are required ie
; N_CIRC=[0.25, 0.5, 0.75, 1.0].
; E_CIRC: Set this keyword to a structure that will contain
; any keywords used in the PLOTS procedure which draws
; the circles. Use this to control the color, thicknes
; and linestyle of the circles. The defaults for these
; are the same as for the PLOTS procedure.
;
; COPY_FIRST: Set this keyword to copy the the first array
; element of the input arrays VAR and WIND_DIRN to the
; end of these arrays. This is to "close" the wind
; rose line.
;
; NODATA: Set this keyword to plot no data. This is useful
; for visualising wind directions for a particuar

```

```

; location by plotting a map in the background with the
; windrose angles or coordinates over the top.
;
;
; EXAMPLE:
; To plot a wind rose of the variable ch4conc with a map
; centred over Cape Grim, with scale of 10e6, no grid lines
; on the map, 4 circles, cross hairs and the default
; baseline indicators:
; PLOT_WIND_ROSE, ch4conc, wind_dir, $
; color=pen(2), thick=6, linestyle=0, $
; title='Methane at Cape Grim', $
; ytitle='CH!D4!N (ppb)', /cross_hairs, $
; map=1, E_MAP={scale:10e6}, $
; E_cont={thick:4,color:pen(4)}, $
; /baseline, $
; E_BASE={thick:4, color:pen(3), linestyle:2}, $
; E_CIRC={thick:2}, n_circ=[0.25,0.5,0.75,1.0]
;
; To plot the above with filled continents and a grid
; on the map:
; PLOT_WIND_ROSE, ch4conc, wind_dir, $
; color=pen(2), thick=6, linestyle=0, $
; title='Methane at Cape Grim', $
; ytitle='CH!D4!N (ppb)', /cross_hairs, $
; map=1, E_MAP={scale:10e6}, $
; E_cont={thick:4,color:pen(4),fill_continents:1}, $
; /baseline, $
; E_BASE={thick:4, color:pen(3), linestyle:2}, $
; E_CIRC={thick:2}, n_circ=[0.25,0.5,0.75,1.0], $
; /GRID, E_GRID={thick:0.5}
;
; To plot just a simple wind rose:
; PLOT_WIND_ROSE, ch4conc, wind_dir, ytitle='CH!D4!N (ppb)'
;
; MODIFICATION HISTORY:
; Written by: Bronwyn Dunse and Paul Krummel,
; 27 October 1998, CSIRO Atmospheric Research, Australia.
; Modified by: Paul Krummel, 8 November 1998. Added map
; functionality.
; Modified by: Paul Krummel, 15 January 1999. Added many
; extra keywords (E_???), cleaned up the routine, added
; help and proper header information, also other small
; improvements and bug fixes.
; MODIFIED by: Paul Krummel, 21 January 1999. Fully documented.
; MODIFIED by: Paul Krummel, 10 February 1999. Added
; NODATA keyword.
; Modified by: Paul Krummel, 11 May 1999. Fixed Title problem
; and added COPY_FIRST keyword.

```



```

max_var = (n_elements(var2) gt 0) ? max([max_var,var2]) : max_var
if keyword_set(nodata) then var[*]=0.0
;
; If MAXV keyword is set then use its value for max_var.
if n_elements(Maxv) eq 1 then max_var=Maxv
;
; If the COPY_FIRST keyword is set then copy the first
; array element of VAR and WIND_DIRN to the end of these
; arrays.
if keyword_set(Copy_First) then begin
var=[var,var[0]] & wind_dirn=[wind_dirn,wind_dirn[0]]
if n_elements(var2) gt 0 then begin
var2=[var2,var2[0]] & wind_dirn2=[wind_dirn2,wind_dirn2[0]]
endif
endif
endif
;
; Check if the N_Rad_Lines keyword was set, if not default it to 12.
N_Rad_Lines = n_elements(N_Rad_Lines) gt 0 ? N_Rad_Lines : 12
;
; +++++
; Setup the plot coordinates. Make the plot isotropic and
; set the x and y range to +/- the maximum value.
PLOT, var, (450.-wind_dirn)*!DTOR, /nodata, /noerase,$
/polar, /isotropic,$
xstyle=5, ystyle=5, $
yrange=[-max_var,max_var], xrange=[-max_var,max_var]

; +++++
; If the MAP keyword is selected then plot a map behind
; the wind rose centred on the given coordinates. If no
; coordinates are given use lat and lon for Cape Grim,
; Tasmania, Australia.
if keyword_set(MAP) then begin

; Check if there is more than one element in the map
; keyword, if so it should contain the lat and lon
; for the map centre. Set this accordingly.
if n_elements(map) gt 1 then begin
lat=map[0] & lon=map[1]
endif else begin ; else set the default map centre
; to Cape Grim.
lat=-40.6829441667D & lon=144.689568611D
endelse

; Check how many plots per page are to be made.
; Use different code for just one plot or more
; than one plot and adjust !p.multi accordingly.
if !p.multi[1] gt 1 or !p.multi[2] gt 1 then begin

```

```

;+ Code for more than one plot per page +
num_left=!p.multi[0]
if num_left eq 0 then num_left=!p.multi[1]*!p.multi[2]

; Setup the mapping here, forcing the map
; into the plot coordinates set above.
map_set, lat, lon, scale=5.e6, /merc, /noborder, $
position=[!x.window[0],!y.window[0], $
!x.window[1],!y.window[1]], $
/advance, _EXTRA=E_MAP

; Plot the coastline here, NOTE the default
; is to use hires map.
map_continents, /coasts, /hires, _EXTRA=E_CONT

; Plot a grid on the map if requested.
if KEYWORD_SET(GRID) then map_grid, _EXTRA=E_GRID

; reset !p.multi to stop frame/page advance.
!p.multi[0]=num_left

endif else begin
;+ Code for one plot per page +

; Setup the mapping here, forcing the map
; into the plot coordinates set above.
map_set, lat, lon, /merc, scale=5.e6, /noborder, $
position=[!x.window[0],!y.window[0], $
!x.window[1],!y.window[1]], $
_EXTRA=E_MAP

; Plot the coastline here, NOTE the default
; is to use hires map.
map_continents, /coasts, /hires, _EXTRA=E_CONT

; Plot a grid on the map if requested.
if KEYWORD_SET(GRID) then map_grid, _EXTRA=E_GRID

; reset !p.multi to stop frame/page advance.
if !p.multi[0] le 0 then !p.multi[0]=1

endelse

endif ; End of mapping section.

; +++++
; Setup the plot coordinates again and get the tickmark values.

```

```

; Set the title and add !C (new line) to the end of it.
Title = keyword_set(Title) ? Title+'!C ' : Title
PLOT, var, (450.-wind_dirn)*!DTOR, /nodata, $
/polar, /isotropic, $
TITLE=Title, $
xstyle=5, ystyle=5, $
yrange=[-max_var,max_var],xrange=[-max_var,max_var],$
ytick_get=tick_val

; ++++
; Plot the circles here.
; Default circles at 0.5 and 1 of Max_var.
if not KEYWORD_SET(n_circ) then n_circ=[0.5,1.0]
for i=0,n_elements(n_circ)-1 do $
  PlotS, circle(0, 0, n_circ[i]*Max_var, nPts=361), _EXTRA=E_CIRC

; ++++
; Count the number of tickmarks
ntick_val=n_elements(tick_val)

; Set the negative tickmarks to positive, for use in labelling.
ptick_val=abs(tick_val)

; Set the default thickness and number of minor tick marks
; if they were not defined in the calling routine.
ythick = keyword_set(ythick) ? ythick : 2
yminor = keyword_set(yminor) ? yminor : 5

; ++++
; Plot a y-axis to the left of the rose with the tick mark values
; and optional ytitle. Default thickness and minor tickmarks are
; used unless otherwise specified.
AXIS,-(max_var+0.22*max_var),0, YAXIS=0, YTHICK=YThick, $
YTICKNAME=format_axis_values(ptick_val), $
YTICKS=ntick_val-1, YTICKV=tick_val, $
YTITLE=YTitle, YMINOR=YMinor, $
YSTYLE=1, YRANGE=[-max_var,max_var]

; ++++
; CROSS HAIRS:
; Place tick marks on the horizontal and vertical radial
; lines through the centre of the plot if the CROSS_HAIRS
; keyword is set.
if KEYWORD_SET(Cross_hairs) then begin
  AXIS, 0, 0, YAXIS=0, YTHICK=0.5,$
  YTICKFORMAT='(a1)', $
  YTICKS=ntick_val-1, YTICKV=tick_val, $
  YMINOR=5, YSTYLE=1, $

```

```

YRANGE=[-max_var,max_var], $
_EXTRA=E_Cross

AXIS, 0, 0, YAXIS=1, YTHICK=0.5,$
YTICKFORMAT='(a1)', $
YTICKS=ntick_val-1, YTICKV=tick_val, $
YMINOR=5, YSTYLE=1, $
YRANGE=[-max_var,max_var], $
_EXTRA=E_Cross

AXIS, 0, 0, XAXIS=0, XTHICK=0.5,$
XTICKFORMAT='(a1)', $
XTICKS=ntick_val-1, XTICKV=tick_val, $
XMINOR=5, XSTYLE=1, $
XRANGE=[-max_var,max_var], $
_EXTRA=E_Cross

AXIS, 0, 0, XAXIS=1, XTHICK=0.5,$
XTICKFORMAT='(a1)', $
XTICKS=ntick_val-1, XTICKV=tick_val, $
XMINOR=5, XSTYLE=1, $
XRANGE=[-max_var,max_var], $
_EXTRA=E_Cross

; If NO radial lines are requested but crosshairs are, then
; label the angles for the four cross hairs. Just set
; N_Rad_lines equal to 4.
if n_elements(N_Rad_Lines) gt 0 and N_Rad_Lines eq 0 then N_Rad_Lines=4

endif

; ++++
; Plot the radial lines - this is now a more general way of doing this,
; allows for easier changing of the number of radial lines. PBK 22 Feb.
2000.
;
; First check if actually want to plot radial lines.
if n_elements(N_Rad_Lines) gt 0 and N_Rad_Lines gt 0 then begin
;
; Check to see if the N_RAD_LINES keyword is set, if so
; then set the number of radial lines to N_RAD_LINES else
; default to 12. NO NEED TO DO THIS NOW< IS DONE ABOVE - PBK 25 July 2000.
;n_rl = n_elements(N_Rad_Lines) gt 0 ? N_Rad_Lines : 12
n_rl = N_Rad_Lines
;
; Calculate radial line spacing.
rl_spac=360./float(n_rl)
a=replicate(max_var,n_rl)

```

```

; Calculate plot angles, l_theta.
l_theta=!DTOR*(450.-rl_spac*findgen(n_rl))
; Plot the lines
for i=0,n_rl-1 do OPLOT, [0,a[i]], [0,l_theta[i]], /polar, $
    thick=1.0, _EXTRA=E_Rad_Lines
;
; ++++
; Annotate the radial lines - this is now a more general way of doing this,
; allows for easier changing of the number of radial lines. PBK 22 Feb.
2000.
; Setup the radius values for the labels, including an offset.
l_radius=a+max_var*0.035-abs(cos(l_theta)*0.02*max_var)

; Turn the radius and angle into rectangular coords. Use theta from above.
polar_coord=make_array(2,n_rl,/float)
polar_coord[0,*]=l_theta & polar_coord[1,*]=l_radius
result=cv_coord(from_polar=polar_coord,/to_rect)

; Set the label string here. Now more general.
label=strcompress(fix(rl_spac*indgen(n_rl)),/remove_all)

; Assign the rectangular coords. Adjust y value.
l_x=result[0,*]
l_y=result[1,*]-max_var*0.02

; Set the alignment for xyouts.
l_align=replicate(0.5,n_rl)
th_hold=fix(450.0-l_theta*!RADEG)
al0=where(th_hold gt 0 and th_hold lt 180, cnt_al0)
if cnt_al0 gt 0 then l_align[al0]=0.
al1=where(th_hold lt 360 and th_hold gt 180, cnt_al1)
if cnt_al1 gt 0 then l_align[al1]=1.

; ++++
; Try to set the character size for the labels according
; to the number of plots per page.
nplots=float(total(!p.multi[1:2]))
case 1 of
nplots le 2.: chsize=1.
nplots gt 2. and nplots le 4.: chsize=0.65
nplots gt 4. and nplots le 6.: chsize=0.5
nplots gt 6.: chsize=0.4
endcase

; Plot the labels.
for i=0,n_rl-1 do XYOUTS, l_x[i], l_y[i], label[i], /data, $
    CHARSIZE=chsize, align=l_align[i]
;

```

```

; endif for plotting radial lines
endif
; +++++
; Plot two radial lines indicating the baseline sector
; if the BASELINE keyword is set. Can also be used to
; plot any type of indicating line.
if KEYWORD_SET(Baseline) then begin
; If just the baseline keyword is set then plot the
; default baseline indicators for Cape Grim, ie 190
; and 280 degrees. If there are two elements or more
; in Baseline keyword, they are angles so plot lines
; along these angles.
if n_elements(Baseline) eq 1 then b_theta=[190.,280.] $
else b_theta=baseline

; Set the radius to plot out to and convert angles to
; compass coords and radians.
b_radius=max_var
b_theta=(450.-b_theta)*!DTOR

; Plot the lines here.
for i=0,n_elements(b_theta)-1 do $
  OPLOT,[0,b_radius],[0,b_theta[i]],/polar, _EXTRA=E_Base
endif

; +++++
; Plot the actual data here! Turn the winddirn into
; radians and offset it appropriately for plotting
; in compass coordinate system (North-South, East-West).
linestyle = keyword_set(linestyle) ? linestyle : 0
thick = keyword_set(thick) ? thick : 4
color = keyword_set(color) ? color : 0

OPLOT, var, (450.-wind_dirn)*!DTOR, /polar, $
linestyle=linestyle, thick=thick, color=color
;
if n_elements(var2) gt 0 then begin
v2_linestyle = keyword_set(v2_linestyle) ? v2_linestyle : 1
v2_thick = keyword_set(v2_thick) ? v2_thick : 4
v2_color = keyword_set(v2_color) ? v2_color : 0
OPLOT, var2, (450.-wind_dirn2)*!DTOR, /polar, $
linestyle=v2_linestyle, thick=v2_thick, color=v2_color
endif
;
; +++++

END

```