
Subject: Reversed log axes in object graphics

Posted by [Randall Skelton](#) on Mon, 18 Oct 2004 17:10:05 GMT

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

Firstly, thanks for reading. I realize with a subject line as above, many will be scared away...

I am trying to move one particular class of plotting routines (that I use very frequently) to object graphics and I've run into a problem. Essentially, I cannot plot a reversed, logarithmic axis. Below is some code which shows my problem. The equivalent direct graphics code as per David's tip (http://www.dfanning.com/graphics_tips/yadayada.html) is shown to illustrate what I expect the plot to look like. All contributions greatly appreciated!

Cheers,
Randall

```
=== log_test.pro ===
```

```
pro log_test
```

```
compile_opt DEFINT32
compile_opt STRICTARR
compile_opt STRICTARRSUBS
compile_opt LOGICAL_PREDICATE
```

```
p = [1013.00, 898.800, 795.000, 701.200, 616.600,
540.500, $
472.200, 411.100, 356.500, 308.000, 265.000,
227.000, $
194.000, 165.800, 141.700, 121.100, 103.500,
88.5000, $
75.6500, 64.6700, 55.2900, 47.2900, 40.4700,
34.6700, $
29.7200, 25.4900, 17.4300, 11.9700, 8.01000,
5.74600, $
4.15000, 2.87100, 2.06000, 1.49100, 1.09000,
0.797800, $
0.425000, 0.219000, 0.109000, 0.0522000, 0.0240000,
0.0105000, $
0.00446000, 0.00184000, 0.000760000, 0.000320000, 0.000145000,
7.10000e-05, $
4.01000e-05, 2.54000e-05 ]
```

```
t = [288.200, 281.700, 275.200, 268.700, 262.200, 255.700, 249.200, $
242.700, 236.200, 229.700, 223.300, 216.800, 216.700,
216.700, $
```

```
216.700, 216.700, 216.700, 216.700, 216.700, 216.700,
216.700, $
217.600, 218.600, 219.600, 220.600, 221.600, 224.000,
226.500, $
230.000, 236.500, 242.900, 250.400, 257.300, 264.200,
270.600, $
270.700, 260.800, 247.000, 233.300, 219.600, 208.400,
198.600, $
188.900, 186.900, 188.400, 195.100, 208.800, 240.000,
300.000, $
360.00 ]
```

```
z = [0.00000, 1.00000, 2.00000, 3.00000, 4.00000, 5.00000, 6.00000, $
7.00000, 8.00000, 9.00000, 10.0000, 11.0000, 12.0000,
13.0000, $
14.0000, 15.0000, 16.0000, 17.0000, 18.0000, 19.0000,
20.0000, $
21.0000, 22.0000, 23.0000, 24.0000, 25.0000, 27.5000,
30.0000, $
32.5000, 35.0000, 37.5000, 40.0000, 42.5000, 45.0000,
47.5000, $
50.0000, 55.0000, 60.0000, 65.0000, 70.0000, 75.0000,
80.0000, $
85.0000, 90.0000, 95.0000, 100.000, 105.000, 110.000,
115.000, $
120.000 ]
```

```
sp = size(p)
nlev = sp[1]
```

```
xtitle = 'Temperature (K)'
ytitle = 'Pressure (mbar)'
ytitle2 = 'Altitude (km)'
```

```
xrange = [min(t),max(t)]
yrange = [p[0], p[nlev-1]]
yrange2 = [min(z), max(z)]
```

```
;;; Plot in direct graphics (4 lines)
!x.margin=[10,8]
plot_io, ystyle=9, t, p, xtitle=xtitle, ytitle=ytitle,$
yrange=yrange, xrange=xrange, xstyle=1
```

```
idx = where ( z mod 10 eq 0 )
axis, yaxis=1, yticks=12, ytickv=p[idx], $
ytickname=string(z[idx],format='(i3)'), ytitle=ytitle2
```

```
;;; Plot in object graphics (more than 4 lines)
```

; Create view object.

```
thisView = OBJ_NEW('IDLgrView', Viewplane_Rect=[-1.0,-1.0,2,2], $  
ZClip=[1.15,-1.15] )
```

; Create model object.

```
thisModel = OBJ_NEW('IDLgrModel')  
thisView->Add, thisModel
```

; Create font object.

```
helvetica10pt = Obj_New('IDLgrFont', 'Helvetica', Size=10)
```

; Create helper objects. First, create title objects
; for the axes and plot. Color them green. Title objects
; will need to be located by hand for reversible axes.
; Be sure you DON'T add the title objects directly to
; the axis objects.

```
xTitleObj = Obj_New('IDLgrText', xtitle, $  
Font=helvetica10pt, /Enable_Formatting)  
yTitleObj = Obj_New('IDLgrText', ytitle, $  
Font=helvetica10pt, /Enable_Formatting)  
yTitleObj2 = Obj_New('IDLgrText', ytitle2, $  
Font=helvetica10pt, /Enable_Formatting)
```

; Create a plot

```
thisPlot = OBJ_NEW('IDLgrPlot', t, p, _Extra=extra)
```

; Get the data ranges of the plot.

```
thisPlot->GetProperty, XRange=xrange, YRange=yrange
```

```
ticklen = 0.02
```

; Axes are created after the plot so the range can be
; set correctly.

```
xAxis = Obj_New("IDLgrAxis", 0, Ticklen=ticklen, $  
Minor=4, Range=xrange, /Extend)  
xAxis->GetProperty, Ticktext=xAxisText  
xAxisText->SetProperty, Font=helvetica10pt
```

```
xAxis2 = Obj_New("IDLgrAxis", 0, Ticklen=ticklen, $  
Minor=4, Range=xrange, /Extend, TICKDIR=1, TEXTPOS=1, NOTEXT=1)
```

```
yAxis = Obj_New("IDLgrAxis", 1, Ticklen=ticklen, $
Minor=4, Range=reverse(yrange), /Log, /Extend)
yAxis->GetProperty, Ticktext=yAxisText
yAxisText->SetProperty, Font=helvetica10pt
```

```
yAxis2 = Obj_New("IDLgrAxis", 1, Ticklen=ticklen, $
Minor=4, Range=yrange2, /Extend, TICKDIR=1, TEXTPOS=1)
yAxis2->GetProperty, Ticktext=yAxisText2
yAxisText2->SetProperty, Font=helvetica10pt
```

```
; The axes may not use exact axis scaling, so the ranges may
; have changed from what they were originally set to. Get
; and update the range variables.
```

```
xAxis->GetProperty, CRange=xrange
xAxis2->GetProperty, CRange=xrange2
yAxis->GetProperty, CRange=yrange
yAxis2->GetProperty, CRange=yrange2
```

```
xs = Normalize(xrange, Position=[-0.5,0.5])
xs2 = Normalize(xrange2, Position=[-0.5,0.5])
ys = Normalize(yrange, Position=[-0.5,0.5])
ys2 = Normalize(yrange2, Position=[-0.5,0.5])
```

```
; Scale the axes and place them in the coordinate space.
; Note that not all values in the Location keyword are
; used. (I've put really large values into the positions
; that are not being used to demonstrate this.) For
; example, with the X axis only the Y and Z locations are used.
```

```
xAxis->SetProperty, Location=[9999.0, -0.5, -0.5], XCoord_Conv=xs
xAxis2->SetProperty, Location=[9999.0, 0.5, -0.5], XCoord_Conv=xs2
yAxis->SetProperty, Location=[-0.5, 9999.0, -0.5], YCoord_Conv=ys
yAxis2->SetProperty, Location=[0.5, 9999.0, -0.5], YCoord_Conv=ys2
```

```
; Scale the Plot.
```

```
thisPlot->SetProperty, XCoord_Conv=xs, YCoord_Conv=ys
```

```
; Add the plot and axes objects to the model.
```

```
thisModel->Add, thisPlot
thisModel->Add, xAxis
thisModel->Add, xAxis2
thisModel->Add, yAxis
thisModel->Add, yAxis2
```

; Add the title objects to the model.

```
thisModel->Add, xTitleObj  
thisModel->Add, yTitleObj  
thisModel->Add, yTitleObj2
```

; Get the window destination object, which is the value of
; an object draw widget. The view will be drawn in the window
; when the window is exposed.

```
thisWindow = obj_new('IDLgrWindow', DIMENSION=[15,15], UNITS=2, $  
GRAPHICS_TREE=thisView)  
thisWindow -> erase
```

; Once we have a window, find the size of the character box
surrounding the
; axis annotation, and calculate a location for the axis titles.
Note that the Y
; dimension of the X axis text box is always about 75% of what it
should be. This
; is the reason the X axis title always appears too close to the
axis compared
; to the Y and Z axis in the normal default placement. That is why
you see that
; number multiplied by 1.5 for the XTitleObj below. (The values
-0.5, 0.5, and 0
; are the endpoints and middle, respectively, of my axis in my
viewport rectangle.)

; To orient the text properly, you must supply the proper baseline
and up direction
; vectors to the Y and Z titles. The X title does not need this,
since the X "situation"
; is the default case. For example, read the Y title orientation
like this: draw the
; text parallel to the Y axis (Baseline=[0,1,0]), with the up
direction in the -X direction
; (UpDir=[-1,0,0]).

```
d = thisWindow->GetTextDimensions(xAxisText)  
xTitleObj->SetProperty, Location=[0, -0.5 - d[1]*2-ticklen, -0.5], $  
Alignment=0.5
```

```
d = thisWindow->GetTextDimensions(yAxisText)  
yTitleObj->SetProperty, Location=[-0.5 - d[0]-2*ticklen, 0, -0.5], $  
Baseline=[0,1,0], UpDir=[-1,0,0], Alignment=0.5
```

```
d = thisWindow->GetTextDimensions(yAxisText2)
```

```
yTitleObj2->SetProperty, Location=[0.5 + d[0]*2-ticklen, 0, -0.5], $  
Baseline=[0,1,0], UpDir=[-1,0,0], Alignment=0.5  
; Draw the window
```

```
thisWindow -> draw
```

```
end
```

Subject: Re: Reversed log axes in object graphics
Posted by [David Fanning](#) on Wed, 20 Oct 2004 11:14:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

Christopher Lee writes:

> I'm sure I read somewhere that you have to make your own log axis in object
> graphics. Anyway, the log_test_new I have pasted below works fine.

Well, as I say, unbelievably straightforward. :-)

Cheers,

David

--

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

Subject: Re: Reversed log axes in object graphics
Posted by [Paul Van Delst\[1\]](#) on Wed, 20 Oct 2004 13:59:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

Christopher Lee wrote:

> In article <1098146457.937748.74790@c13g2000cwb.googlegroups.com>,
> "Randall Skelton" <randall.skelton@gmail.com> wrote:

>

> . think of it as good-morning challenge :-)

>

>> Thanks in advance,

>> Randall

>>

>

>

> I'm sure I read somewhere that you have to make your own log axis in object

- > graphics. Anyway, the log_test_new I have pasted below works fine.
- >
- > Steps to make it work (ones that I can remember, for everything else,
- > there's diff.)
- >
- > 1. change the pressure coordinate to alog10 (pressure)
- > 2. calculate the log10 ranges for your data and log10 tick values
- > 3. calculate the 'normal' ticknames for use in 'ticktext' (The ticknames
- > formatl used are quite bad, you can obviously use any
- > names/format you want here)
- > 4. Define a ticktext object for yaxis, instead of leaving IDLgrAxis
- > define it (you need to delete it now), removed the /Log keyword.
- > 5. negate the yscale values returned from Normalize (this may be a fix
- > for a bug I introduced somewhere, but It works)
- > 6. I think that was it.

[curmudgeon alert!]

My goodness. Apart from the complexity, what an example of code bloat! I assume this is IDL specific since I cannot imagine people designing other languages think something like this is an improvement over the 3-line direct graphics version.

This thread is a good case study as to why people don't (or maybe shouldn't?) use object graphics.

paulv

p.s. Just to be clear, my misgivings aren't directed towards you or the solution you provided, but the need for it in the first place.

- >
- > I was feeling lucky, so I also changed yaxis to yaxis_major and added a
- > yaxis_minor to plot the minor ticks, in this case, Minor =0 for both
- > new axis, and the /notext keyword is used yaxis_minor.
- >
- > Chris.
- >
- >
- > ;;--
- > pro log_test_new
- >
- > compile_opt DEFINT32
- > compile_opt STRICTARR
- > compile_opt STRICTARRSUBS
- > compile_opt LOGICAL_PREDICATE
- >
- > p = [1013.00, 898.800, 795.000, 701.200, 616.600, 540.500, \$

```

> 472.200, 411.100, 356.500, 308.000, 265.000,227.000, $
> 194.000, 165.800, 141.700, 121.100, 103.500,88.5000, $
> 75.6500, 64.6700, 55.2900, 47.2900, 40.4700,34.6700, $
> 29.7200, 25.4900, 17.4300, 11.9700, 8.01000,5.74600, $
> 4.15000, 2.87100, 2.06000, 1.49100, 1.09000,0.797800, $
> 0.425000, 0.219000, 0.109000, 0.0522000, 0.0240000,0.0105000, $
> 0.00446000, 0.00184000, 0.000760000, 0.000320000, 0.000145000,7.10000e-05, $
> 4.01000e-05, 2.54000e-05 ]
>
> p=alog10(p)
>
> t = [288.200, 281.700, 275.200, 268.700, 262.200, 255.700, 249.200, $
> 242.700, 236.200, 229.700, 223.300, 216.800, 216.700,216.700, $
> 216.700, 216.700, 216.700, 216.700, 216.700, 216.700,216.700, $
> 217.600, 218.600, 219.600, 220.600, 221.600, 224.000,226.500, $
> 230.000, 236.500, 242.900, 250.400, 257.300, 264.200,270.600, $
> 270.700, 260.800, 247.000, 233.300, 219.600, 208.400,198.600, $
> 188.900, 186.900, 188.400, 195.100, 208.800, 240.000,300.000, $
> 360.00 ]
>
> z = [0.00000, 1.00000, 2.00000, 3.00000, 4.00000, 5.00000, 6.00000, $
> 7.00000, 8.00000, 9.00000, 10.0000, 11.0000, 12.0000,13.0000, $
> 14.0000, 15.0000, 16.0000, 17.0000, 18.0000, 19.0000,20.0000, $
> 21.0000, 22.0000, 23.0000, 24.0000, 25.0000, 27.5000,30.0000, $
> 32.5000, 35.0000, 37.5000, 40.0000, 42.5000, 45.0000,47.5000, $
> 50.0000, 55.0000, 60.0000, 65.0000, 70.0000, 75.0000,80.0000, $
> 85.0000, 90.0000, 95.0000, 100.000, 105.000, 110.000,115.000, $
> 120.000 ]
>
> sp = size(p)
> nlev = sp[1]
>
> xtitle = 'Temperature (K)'
> ytitle = 'Pressure (mbar)'
> ytitle2 = 'Altitude (km)'
>
> xrange = [min(t),max(t)]
> yrange = [p[0], p[nlev-1]]
> yrange2 = [min(z), max(z)]
>
> ;;; Plot in direct graphics (4 lines)
> !x.margin=[10,8]
> ;plot_io, ystyle=9, t, 10^p, xtitle=xtitle, ytitle=ytitle,$
> ;yrange=10^yrange, xrange=xrange, xstyle=1
> ;idx = where ( z mod 10 eq 0 )
> ;axis, yaxis=1, yticks=12, ytickv=10^p[idx], $
> ;ytickname=string(z[idx],format='(i3)'), ytitle=ytitle2
>

```

```

> ;; Plot in object graphics (more than 4 lines)
>
> thisView = OBJ_NEW('IDLgrView', Viewplane_Rect=[-1.0,-1.0,2,2], $
> ZClip=[1.15,-1.15] )
>
> thisModel = OBJ_NEW('IDLgrModel')
> thisView->Add, thisModel
>
> helvetica10pt = Obj_New('IDLgrFont', 'Helvetica', Size=10)
> ; Create helper objects. First, create title objects
> ; for the axes and plot. Color them green. Title objects
> ; will need to be located by hand for reversible axes.
> ; Be sure you DON'T add the title objects directly to
> ; the axis objects.
>
> xTitleObj = Obj_New('IDLgrText', xtitle, $
> Font=helvetica10pt, /Enable_Formatting)
> yTitleObj = Obj_New('IDLgrText', ytitle, $
> Font=helvetica10pt, /Enable_Formatting)
> yTitleObj2 = Obj_New('IDLgrText', ytitle2, $
> Font=helvetica10pt, /Enable_Formatting)
>
> ; Create a plot
>
> thisPlot = OBJ_NEW('IDLgrPlot', t, p, _Extra=extra)
>
> ; Get the data ranges of the plot.
> thisPlot->GetProperty, XRange=xrange, YRange=yrange
>
> ticklen = 0.02
>
> ; Axes are created after the plot so the range can be
> ; set correctly.
>
> xAxis = Obj_New("IDLgrAxis", 0, Ticklen=ticklen, $
> Minor=4, Range=xrange, /Extend)
> xAxis->GetProperty, Ticktext=xAxisText
> xAxisText->SetProperty, Font=helvetica10pt
>
> xAxis2 = Obj_New("IDLgrAxis", 0, Ticklen=ticklen, $
> Minor=4, Range=xrange, /Extend, TICKDIR=1, TEXTPOS=1, NOTEXT=1)
>
> ;yAxis = Obj_New("IDLgrAxis", 1, Ticklen=ticklen, $
> ;Minor=4, Range=reverse(yrange), /Log, /Extend)
>
> ;major
> major_ticks=[1]
> n_major=n_elements(major_ticks)

```

```

> major_tick_values=alog10(major_ticks)
> min_decades=floor(min(p))
> n_decades=ceil(max(p))-floor(min(p))
>
> major_tickv=reform( $
>     ((major_tick_values # replicate(1,n_decades)) + $
>     (min_decades+findgen(n_decades)##replicate(1,n_major))), $
>     n_decades*n_major)
>
> major_tickn=reform(10^major_tickv,      $
>     n_decades*n_major)
>
> ;minor
> minor_ticks=[1,2,3,4,5,6,7,8,9]
> n_minor=n_elements(minor_ticks)
> minor_tick_values=alog10(minor_ticks)
> min_decades=floor(min(p))
> n_decades=ceil(max(p))-floor(min(p))
>
> minor_tickv=reform( $
>     ((minor_tick_values # replicate(1,n_decades)) + $
>     (min_decades+findgen(n_decades)##replicate(1,n_minor))), $
>     n_decades*n_minor)
>
> minor_tickn=reform(10^minor_tickv,      $
>     n_decades*n_minor)
> ;---done--
>
>
> yAxisText=Obj_New("IDLgrText" ,string(major_tickn),Font=helvetica10pt,strings=string(major
_tickn,format='(e10.2)'))
>
> yAxis_major = Obj_New("IDLgrAxis", 1, Ticklen=ticklen, $
>     Minor=0, Range=reverse(yrange), /Extend,$
>     tickvalues=major_tickv, ticktext=yAxisText)
>
> yAxis_major->GetProperty,yrange=yr
>
> yAxis_minor = Obj_New("IDLgrAxis", 1, Ticklen=ticklen*1, $
>     Minor=0, Range=reverse(yrange), /Extend,$
>     tickvalues=minor_tickv, /notext)
>
>
> yAxis2 = Obj_New("IDLgrAxis", 1, Ticklen=ticklen, $
>     Minor=4, Range=yrange2, /Extend, TICKDIR=1, TEXTPOS=1)
> yAxis2->GetProperty, Ticktext=yAxisText2
> yAxisText2->SetProperty, Font=helvetica10pt
>

```

```

> ; The axes may not use exact axis scaling, so the ranges may
> ; have changed from what they were originally set to. Get
> ; and update the range variables.
>
> xAxis->GetProperty, CRange=xrange
> xAxis2->GetProperty, CRange=xrange2
> yAxis_major->GetProperty, CRange=yrange_major
> yAxis_minor->GetProperty, CRange=yrange_minor
> yAxis2->GetProperty, CRange=yrange2
>
> xs = Normalize(xrange, Position=[-0.5,0.5])
> xs2 = Normalize(xrange2, Position=[-0.5,0.5])
> ys_major = Normalize(yrange_major, Position=[-0.5,0.5])
> ys_minor = Normalize(yrange_minor, Position=[-0.5,0.5])
> ys2 = Normalize(yrange2, Position=[-0.5,0.5])
>
> ; Scale the axes and place them in the coordinate space.
> ; Note that not all values in the Location keyword are
> ; used. (I've put really large values into the positions
> ; that are not being used to demonstrate this.) For
> ; example, with the X axis only the Y and Z locations are used.
> ys_major=-ys_major
> ys_minor=-ys_minor
>
> xAxis->SetProperty, Location=[9999.0, -0.5, -0.5], XCoord_Conv=xs
> xAxis2->SetProperty, Location=[9999.0, 0.5, -0.5], XCoord_Conv=xs2
> yAxis_major->SetProperty, Location=[-0.5, 9999.0, -0.5],
YCoord_Conv=ys_major,textupdir=[0,-1,0]
> yAxis_minor->SetProperty, Location=[-0.5, 9999.0, -0.5],
YCoord_Conv=ys_minor,textupdir=[0,-1,0]
> yAxis2->SetProperty, Location=[0.5, 9999.0, -0.5], YCoord_Conv=ys2
>
> ; Scale the Plot.
>
> thisPlot->SetProperty, XCoord_Conv=xs, YCoord_Conv=ys_major
>
> ; Add the plot and axes objects to the model.
>
> thisModel->Add, thisPlot
> thisModel->Add, xAxis
> thisModel->Add, xAxis2
> thisModel->Add, yAxis_major
> thisModel->Add, yAxis_minor
> thisModel->Add, yAxis2
>
> ; Add the title objects to the model.
>
> thisModel->Add, xTitleObj

```

```

> thisModel->Add, yTitleObj
> thisModel->Add, yTitleObj2
>
> ; Get the window destination object, which is the value of
> ; an object draw widget. The view will be drawn in the window
> ; when the window is exposed.
>
> thisWindow = obj_new('IDLgrWindow', DIMENSION=[15,15], UNITS=2, $
> GRAPHICS_TREE=thisView)
> thisWindow -> erase
>
> ; Once we have a window, find the size of the character boxes surrounding the
> ; axis annotation, and calculate a location for the axis titles. Note that the Y
> ; dimension of the X axis text box is always about 75% of what it*should* be. This
> ; is the reason the X axis title always appears too close to the axis compared
> ; to the Y and Z axis in the normal default placement. That is why you see that
> ; number multiplied by 1.5 for the XTitleObj below. (The values-0.5, 0.5, and 0
> ; are the endpoints and middle, respectively, of my axis in my viewport rectangle.)
>
> ; To orient the text properly, you must supply the proper baseline and up direction
> ; vectors to the Y and Z titles. The X title does not need this, since the X "situation"
> ; is the default case. For example, read the Y title orientation like this: draw the
> ; text parallel to the Y axis (Baseline=[0,1,0]), with the up direction in the -X direction
> ; (UpDir=[-1,0,0]).
>
> d = thisWindow->GetTextDimensions(xAxisText)
> xTitleObj->SetProperty, Location=[0, -0.5 - d[1]*2-ticklen, -0.5], $
> Alignment=0.5
>
> d = thisWindow->GetTextDimensions(yAxisText)
> yTitleObj->SetProperty, Location=[-0.5 - d[0]-2*ticklen, 0, -0.5], $
> Baseline=[0,1,0], UpDir=[-1,0,0], Alignment=0.5
>
> d = thisWindow->GetTextDimensions(yAxisText2)
> yTitleObj2->SetProperty, Location=[0.5 + d[0]*2-ticklen, 0, -0.5], $
> Baseline=[0,1,0], UpDir=[-1,0,0], Alignment=0.5
> ; Draw the window
>
> thisWindow -> draw
> ;cleanup
> ;obj_destroy, thisWindow
> ;obj_destroy, yaxistext
> ;obj_destroy, helvetica10pt
> ;help, /heap
> end

```

Subject: Re: Reversed log axes in object graphics
Posted by [Randall Skelton](#) on Wed, 20 Oct 2004 16:55:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

Fantastic, thanks a lot Chris! So does anyone know what the /LOG keyword to IDLgrAxis is actually good for?

There are good reasons to want to use object graphics but, in cases like this, I certainly agree that the reasons are hard to find. While being able to control object graphics primitives is a feature, this is one of the worst examples where doing something simple requires a lot of ingenuity.

Thanks again Chris,
Randall

Subject: Re: Reversed log axes in object graphics
Posted by [Chris Lee](#) on Wed, 20 Oct 2004 19:22:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <1098291359.742020.84520@z14g2000cwz.googlegroups.com>, "Randall Skelton" <randall.skelton@gmail.com> wrote:

> Fantastic, thanks a lot Chris! So does anyone know what the /LOG
> keyword to IDLgrAxis is actually good for? There are good reasons to
> want to use object graphics but, in cases like this, I certainly agree
> that the reasons are hard to find. While being able to control object
> graphics primitives is a feature, this is one of the worst examples
> where doing something simple requires a lot of ingenuity.
> Thanks again Chris,
> Randall
>

/LOG does plot a log axis :) no really, within the caveat from David that when you reverse the axis, the text will also be reversed it works fine. Just don't start believing this will affect your plot in any way. The reason for this strange behaviour is simply the linear transformation matrix used by IDL, there's no way to get a logarithmic scaling between data coordinates and device coordinates in a 4x4 matrix.

I'm not sure if I had a "doh!" moment, or an "ah" moment. but version 2 is below, note the use of /Log... 100% genuine log axis.

This time..

1. The data is input into plot as `alog10(real_data)`

2. The yrange is carefully scaled when transferring between data and axes
3. axis text is flipped and corrected as before.

(the function has been reduced to the minimum number of lines possible, I added ";CL" labels to point out the changes.

Chris.

```
pro log_test3
```

```
compile_opt DEFINT32
compile_opt STRICTARR
compile_opt STRICTARRSUBS
compile_opt LOGICAL_PREDICATE
```

```
p = 10^((findgen(100)+1)/10.)
```

```
t = findgen(100)
```

```
sp = size(p)
nlev = sp[1]
```

```
xrange = [min(t),max(t)]
yrange = [p[0], p[nlev-1]]
```

```
; Create view object.
```

```
thisView = OBJ_NEW('IDLgrView', Viewplane_Rect=[-1.0,-1.0,2,2], $
ZClip=[1.15,-1.15] )
```

```
; Create model object.
```

```
thisModel = OBJ_NEW('IDLgrModel')
thisView->Add, thisModel
```

```
; Create a plot
```

```
;CL alog10(p)
thisPlot = OBJ_NEW('IDLgrPlot', t, alog10(p), _Extra=extra)
```

```
; Get the data ranges of the plot.
```

```
thisPlot->GetProperty, XRange=xrange, YRange=yrange
```

```
ticklen = 0.02
;CL fix the yrange
yrange=10^yrange
```

; Axes are created after the plot so the range can be
; set correctly.

```
xAxis = Obj_New("IDLgrAxis", 0, Ticklen=ticklen, $  
Minor=4, Range=xrange, /Extend)
```

```
;CL /LOG keyword  
yAxis = Obj_New("IDLgrAxis", 1, Ticklen=ticklen, $  
Minor=4, Range=yrange,/Log, /Extend)  
print, yrange
```

```
xAxis->GetProperty, CRange=xrange  
yAxis->GetProperty, CRange=yrange
```

```
xs = Normalize(xrange, Position=[-0.5,0.5])  
ys = Normalize(yrange, Position=[-0.5,0.5])
```

```
;CL reverse the axis and flip the text, (textupdir)  
ys=-ys  
xAxis->SetProperty, Location=[9999.0, -0.5, -0.5], XCoord_Conv=xs  
yAxis->SetProperty, Location=[-0.5, 9999.0, -0.5], YCoord_Conv=ys,textupdir=[0,-1,0]
```

; Scale the Plot.

```
thisPlot->SetProperty, XCoord_Conv=xs, YCoord_Conv=ys
```

; Add the plot and axes objects to the model.

```
thisModel->Add, thisPlot  
thisModel->Add, xAxis  
thisModel->Add, yAxis
```

```
thisWindow = obj_new('IDLgrWindow', DIMENSION=[15,15], UNITS=2, $  
GRAPHICS_TREE=thisView)
```

```
thisWindow -> erase  
thisWindow -> draw
```

```
end
```

Subject: Re: Reversed log axes in object graphics
Posted by [Randall Skelton](#) on Wed, 20 Oct 2004 20:17:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

aahhhhhhhhhhhh...

Subject: Re: Reversed log axes in object graphics
Posted by [David Fanning](#) on Wed, 20 Oct 2004 20:24:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

Christopher Lee writes:

> /LOG does plot a log axis :) no really, within the caveat from
> David that when you reverse the axis, the text will also be reversed it
> works fine.

Now this even I can understand. Thanks. :-)

David

--

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

Subject: Re: Reversed log axes in object graphics
Posted by [Chris\[2\]](#) on Thu, 21 Oct 2004 20:09:07 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Randall,

Well, I was able to accomplish your plot using iTools, in about 10 lines of code. It's a bit twisty, but it gets the job done. Plus, it's interactive afterwards!

-Chris
Research Systems, Inc.

```
.***** iPlot axis callback.
```

```
,  
function log_test_format, Axis, Index, Value, DATA=data
```

```
    zvalue = interpol(data.z, data.p, Value)
```

```
    return, string(zvalue, format='(i3)')
```

```
end
```

```
.*****  
,
```

```
p = [1013.00, 898.800, 795.000, 701.200, 616.600, 540.500, $  
472.200, 411.100, 356.500, 308.000, 265.000, 227.000, $  
194.000, 165.800, 141.700, 121.100, 103.500, 88.5000, $  
75.6500, 64.6700, 55.2900, 47.2900, 40.4700, 34.6700, $
```

```

29.7200, 25.4900, 17.4300, 11.9700, 8.01000,5.74600, $
4.15000, 2.87100, 2.06000, 1.49100, 1.09000,0.797800, $
0.425000, 0.219000, 0.109000, 0.0522000, 0.0240000,0.0105000, $
0.00446000, 0.00184000, 0.000760000, 0.000320000, 0.000145000,7.10000e-05, $
4.01000e-05, 2.54000e-05 ]

```

```

t = [288.200, 281.700, 275.200, 268.700, 262.200, 255.700, 249.200, $
242.700, 236.200, 229.700, 223.300, 216.800, 216.700,216.700, $
216.700, 216.700, 216.700, 216.700, 216.700, 216.700,216.700, $
217.600, 218.600, 219.600, 220.600, 221.600, 224.000,226.500, $
230.000, 236.500, 242.900, 250.400, 257.300, 264.200,270.600, $
270.700, 260.800, 247.000, 233.300, 219.600, 208.400,198.600, $
188.900, 186.900, 188.400, 195.100, 208.800, 240.000,300.000, $
360.00 ]

```

```

z = [0.00000, 1.00000, 2.00000, 3.00000, 4.00000, 5.00000, 6.00000, $
7.00000, 8.00000, 9.00000, 10.0000, 11.0000, 12.0000,13.0000, $
14.0000, 15.0000, 16.0000, 17.0000, 18.0000, 19.0000,20.0000, $
21.0000, 22.0000, 23.0000, 24.0000, 25.0000, 27.5000,30.0000, $
32.5000, 35.0000, 37.5000, 40.0000, 42.5000, 45.0000,47.5000, $
50.0000, 55.0000, 60.0000, 65.0000, 70.0000, 75.0000,80.0000, $
85.0000, 90.0000, 95.0000, 100.000, 105.000, 110.000,115.000, $
120.000 ]

```

```

sp = size(p)
nlev = sp[1]

```

```

xtitle = 'Temperature (K)'
ytitle = 'Pressure (mbar)'
ytitle2 = 'Altitude (km)'

```

```

xrange = [min(t),max(t)]
yrange = [p[0], p[nlev-1]]
yrange2 = [min(z), max(z)]

```

```

;;; Plot in direct graphics (4 lines)
!x.margin=[10,8]
plot_io, ystyle=9, t, p, xtitle=xtitle, ytitle=ytitle,$
yrange=yrange, xrange=xrange, xstyle=1

```

```

idx = where ( z mod 10 eq 0 )
axis, yaxis=1, yticks=12, ytickv=p[idx], $
ytickname=string(z[idx],format='(i3)'), ytitle=ytitle2

```

```

;***** Using iTools
,

```

```

iPlot, t, p, /Y_LOG, YRANGE=[MAX(p), MIN(p)], $

```

```

    XTITLE=xtitle, YTITLE=ytitle, YTICKFORMAT='(G0)'
it = ITGETCURRENT(TOOL=oTool)
idAxis = oTool->FindIdentifiers('*Axis3', /VISUALIZATION)
oAxis = oTool->GetByIdentifier(idAxis)
oAxis->SetProperty, NOTEXT=0, AXIS_TITLE=ytitle2, $
    MINOR=0, $
    TICKFORMAT='log_test_format', TICKFRMTDATA={z:z, p:p}
; IDLitVisAxis swallows TICKVALUES, so set directly.
oGrAxis = oAxis->Get(/ALL, ISA='IDLgrAxis')
oGrAxis->SetProperty, TICKVALUES=p[idx]

;*****
;
end

```

"Randall Skelton" <randall.skelton@gmail.com> wrote in message
news:1098119405.276183.192090@f14g2000cwb.googlegroups.com.. .

> Firstly, thanks for reading. I realize with a subject line as above,
> many will be scared away...

>

> I am trying to move one particular class of plotting routines (that I
> use very frequently) to object graphics and I've run into a problem.
> Essentially, I cannot plot a reversed, logarithmic axis. Below is some
> code which shows my problem. The equivalent direct graphics code as per
> David's tip (http://www.dfanning.com/graphics_tips/yadayada.html) is
> shown to illustrate what I expect the plot to look like. All
> contributions greatly appreciated!

>

> Cheers,
> Randall

>

Subject: Re: Reversed log axes in object graphics
Posted by [Randall Skelton](#) on Thu, 21 Oct 2004 23:29:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

Wow. Thanks to all who wrote with ideas and help. I must admit that
I've never been particularly impressed with iTools but between this and
the mapping code posted a few weeks back, it is starting to grow on me.
Is it possible to divert an iTools plot directly to a postscript file
instead of the screen?

Cheers,
Randall

Subject: Re: Reversed log axes in object graphics
Posted by [David Fanning](#) on Fri, 22 Oct 2004 04:50:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

Randall Skelton writes:

- > Wow. Thanks to all who wrote with ideas and help. I must admit that
- > I've never been particularly impressed with iTools but between this and
- > the mapping code posted a few weeks back, it is starting to grow on me.
- > Is it possible to divert an iTools plot directly to a postscript file
- > instead of the screen?

Let's just say I hope your higher hopes survive this. :-(

Cheers,

David

--

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

Subject: Re: Reversed log axes in object graphics
Posted by [David Fanning](#) on Fri, 22 Oct 2004 10:14:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning writes:

- > Let's just say I hope your higher hopes survive this. :-(

I think I may have been too hard on iTools. :-)

I spent a couple of minutes poking around with PostScript output this morning, and I was pleasantly surprised by the quality of the output (I was using iPlot.) I thought it was quite good. I think the little Print Preview interface could use a bit more work (I lost one of the handles when I dragged it outside the window trying to get my plot to fill up the PostScript page), but that is an annoyance rather than a huge limitation. But the representation on the page was very similar to what I was seeing on my display.

I'm still not optimistic about scientists being able to write their own iTools (you can't believe what a struggle

I am having here to introduce a couple of widgets into their lives!), but maybe if enough functionality is provided, they can be useful.

Cheers,

David

P.S. Let's just say I'm not ready to give up my position as Chief Curmudgeon of the IEPA just yet, but maybe in a year or two, when some of the younger members are ready. :-)

--

David W. Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Phone: 970-221-0438, IDL Book Orders: 1-888-461-0155
