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
```