
Subject: Re: IDLgrAxis: how to manipulate tickvalues? and to prevent strange stretching?

Posted by [Karl Schultz](#) on Mon, 06 May 2002 16:23:53 GMT

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

"Harald von der Osten-Woldenburg" <hvdosten@lb.netic.de> wrote in message news:3CD55D43.EDFDE1B5@lb.netic.de...

> Hallo,

>

>

> how can I manipulate tickvalues in IDLgrAxis-objects?

>

>

> What I would like to have is (I have mainly used code by David):

>

snip code fragment

>

> the 3-D data are in the range [8000 <= x <= 20000, 6000 <= y <= 14000]

> but I would like to have them converted (see tickvalues as mentioned

> above). Or have I to convert all the data before reading them into IDL?

> (I don't want to do this)

>

When making plots with Object Graphics, you have to decide at the very start how to set up your coordinate systems.

For very simple plots, many people often change their viewport (VIEWPLANE_RECT prop of IDLgrView) to match their data ranges and then plot away. This works well for simple plots that have only one data range and/or don't vary a lot from one dimension to another. The advantage is that you don't have to scale any of the objects - just fit the view to the data.

But if you want to place more than one plot in the same view, where each plot has a data range that varies from the other plots, or if you want to do something unusual with the axes (like label them differently from the data), then it pays to do "normalized" plotting. Here, you leave the VIEWPLANE_RECT alone (-1 to 1) and scale/translate everything to fit into this range using [XYZ]COORD_CONV.

In the example below, I generate data in the range you want and make a polyline which has vertices in that range. But I use XCOORD_CONV and YCOORD_CONV to scale/translate it into the normalized view. See the suggestions in the docs for XCOORD_CONV to see how to obtain the values to use.

I also did the same thing with the axes, but using different values for

XCOORD_CONV and YCOORD_CONV because the tick values are different from the data.

Note that I'm not actually changing the original data - just applying a transform with [XYZ]COORD_CONV.

Also note that if you want to add other stuff to the view later, it is pretty easy to use [XYZ]COORD_CONV to put new objects in the view without having to worry about adjusting everything else in the view if you should have to enlarge the view to accomodate the new objects. You can layout or plan the placement of your objects in normalized space, which is convenient.

pro axes

```
helvetica8pt = obj_new('IDLgrFont','helvetica', size=8)
helvetica2pt = obj_new('IDLgrFont','helvetica', size=2)
xtitleObj = obj_new('IDLgrText', "X axis")
ytitleObj = obj_new('IDLgrText', "Y axis")
```

```
:: the 3-D data are in the range [8000 <= x <= 20000, 6000 <= y <= 14000]
```

```
xData = RANDOMU(seed, 100) * 12000 + 8000
yData = RANDOMU(seed, 100) * 8000 + 6000
oLine = obj_new('IDLgrPolyline', xData, yData, $
  XCOORD_CONV = [-8000.0/12000, 1/12000.0], YCOORD_CONV = [-6000.0/8000,
  1/8000.0])
```

```
xAxis = obj_new("IDLgrAxis",0, Color=[0,0,0], Ticklen=-0.04, Thick=2.0,$
  Minor=1, Title=xtitleObj, $
  /exact,tickvalues=[-60,-40,-20,-0,20,40,60], $
  XCOORD_CONV = [0.0, 1/60.0])
xAxis -> GetProperty, Ticktext=xAxisText
xAxisText -> SetProperty, Font=helvetica8pt, Recompute_Dimensions=2
```

```
yAxis = obj_new("IDLgrAxis",1, Color=[0,0,0], Ticklen=-0.05, Thick=2.0,$
  Minor=1, Title=ytitleObj, $
  /exact,textpos=0,tickvalues=[-40,-20,0,20,40], $
  YCOORD_CONV = [0.0, 1/40.0])
yAxis-> GetProperty, Ticktext=yAxisText
yAxisText -> SetProperty, Font=helvetica2pt, Recompute_Dimensions=2
```

```
oWin = obj_new('IDLgrWindow')
oView = obj_new('IDLgrView')
oModel = obj_new('IDLgrModel')
oView->Add, oModel
oModel->Add, [xAxis, yAxis, oLine]
oWin->Draw, oView
end
```

This code won't solve all of your problems, but just illustrates another approach that may make things easier to deal with.

Note that I could have gone with shaping the view around the data and then using the [XYZ]COORD_CONV props on either the axes or the data to get it all to work, but it would have been pretty confusing. You'd have to establish a relationship between the data and the tick values and then use the [XYZ]COORD_CONV to fix it up. As a really simple example, you may have data in the range [0, 1000] and want your tick labels to be 1/10th of the data values. If you set the VIEWPLANE_RECT to [0,1000] (I'm only talking about one dimension in this example), then you could set the axis range to [0,100] and XCOORD_CONV to scale it by 10.

Something like:

```
pro axes2
```

```
oAxis = obj_new('IDLgrAxis', 0, range=[0,100], XCOORD_CONV=[0,10])
oLine = obj_new('IDLgrPolyline', [0, 1000], [10,10])
```

```
oWin = obj_new('IDLgrWindow')
oView = obj_new('IDLgrView', VIEWPLANE_RECT=[-100,-50,1500,200])
oModel = obj_new('IDLgrModel')
oView->Add, oModel
oModel->Add, [oAxis, oLine]
oWin->Draw, oView
end
```

```
>
> Second question: How can I prevent the strange stretching of the labels
> and tickvalues of the y-axis? I have changes the size of he font to 2pt,
> and now the labels are similar to the labels of the x-Axis using 8pt.
> But it is not satisfying....
```

I didn't really see this happening, but that may be because I didn't have enough of your code. The view rectangle may have been important here, for example. If you could post a more complete ready-to-run program that shows the problem, we could help more.

But note that this kind of problem may be prevented by using the normalized view. I note that in the first example, the 2pt text is way too small and you could probably go back to 8pt.

This whole topic is one of those religious discussions and there are good reasons for either alternative.

Hope this helps,
Karl
