Subject: Re: plot range in Object Graphics
Posted by davidf on Wed, 10 May 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Brad Gom (b_gom@hotmail.com) writes:

> If I have a simple x-y plot in object-graphics, how to I make the plot
> lines extend all the way to the axis borders like they do in direct
> graphics? For example, the following code produces a line plot that
> extends flush to the right axis:
>
> x=findgen(10)
> plot, x, sin(x), xrange=[0,8.5], /xstyle
>
> If I make an IDLgrPlot object with the same data and xrange, and some
> IDLgrAxis objects with the proper xrange and yrange, then the plot stops
> at the last datapoint- namely x=8.
>
> Is there an obvious way to fix this that I'm missing?

Uh, well, I get identical results in my object graphics
XPLOT program (download the latest, there have been some
changes in most of these object program examples in the
past week or so) as I do in the direct graphics case.
Perhaps you are scaling the axes incorrectly. That is
*easy* to do in object graphics.

```
XPlot, x, sin(x), xrange=[0,8.5], Exact=[1,0]
```

The program is available here:

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

Note that the zip files in this directory have NOT
yet been updated with the program changes from the
past two weeks. Maybe later this week. :-)

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: plot range in Object Graphics
Posted by davidf on Thu, 11 May 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Brad Gom (b_gom@hotmail.com) writes:

> Pavel Romashkin wrote:
>
>> I think you also have to apply the same normalization (xcoord_conv) the
>> rest of the plot has, to the one you are adding.
>
> Right. I didn't mention that.
>
>> In cases like this, I tried also keeping plot limits in an array, to avoid
>> multiple looped
>> queries to IDLgrPlots contained in my graph. This way, it is easy to
>> reset the limits of the axes if, lets say, you remove the plot that
>> caused axes to extend to a certain limit.
>
> Another simple minded approach I thought of was making some IDLgrPolygon objects
> that sit around the plot limits. This way, any lines that were plotted beyond the
> axis would be hidden by the polygons. There would be alot of extra rendering
> happening of course. Despite the inelegance, the advantage would be that you
> wouldn't have to change the ranges of the plots.

I really think this is a simple problem of not scaling
either the data or the axes properly. This is *so* easy
to do in object graphics. In fact, I'm convinced most
people are scaling their data improperly because they don't
realize that object axes autoscale. They only discover their
mistakes when they get down to the fine detail of something
like this. And by then, who knows how many scientists have
been led astray by erroneous illustrations in scientific
journals.

You have to go get the *actual* values of your axes and
scale the data to those, or you will get exactly the problems
you describe.

Cheers,

David

P.S. Please don't implement the Polygon idea. It's just to
grim to imagine and is in the worst tradition of just getting
the damn thing to work. :-(

--
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: plot range in Object Graphics
Posted by Brad Gom on Thu, 11 May 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Pavel Romashkin wrote:

> I think you also have to apply the same normalization (xcoord_conv) the
> rest of the plot has, to the one you are adding.

Right. I didn't mention that.

> In cases like this, I tried also keeping plot limits in an array, to avoid
> multiple looped
> queries to IDLgrPlots contained in my graph. This way, it is easy to
> reset the limits of the axes if, lets say, you remove the plot that
> caused axes to extend to a certain limit.

Another simple minded approach I thought of was making some IDLgrPolygon objects that sit around the plot limits. This way, any lines that were plotted beyond the axis would be hidden by the polygons. There would be alot of extra rendering happening of course. Despite the inelegance, the advantage would be that you wouldn't have to change the ranges of the plots.

Brad

P.S. This illustrates one reason why I wish there was a way to call methods on an object array without looping: when you are setting a property of all the objects to the same value.
( PlotObjArr->SetProperty, xrange=xr,yrange=yr )

Subject: Re: plot range in Object Graphics
Posted by promashkin on Thu, 11 May 2000 07:00:00 GMT
View Forum Message <> Reply to Message

I think you also have to apply the same normalization (xcoord_conv) the rest of the plot has, to the one you are adding. In cases like this, I tried also keeping plot limits in an array, to avoid multiple looped queries to IDLgrPlots contained in my graph. This way, it is easy to reset the limits of the axes if, lets say, you remove the plot that caused axes to extend to a certain limit.

Cheers,
Pavel

Brad Gom wrote:
> Now I want to go one further. In my object I want to add and remove plot lines
> to the model without creating new axes. When initializing my object, I go
> through the steps above. So here's my question:
> Suppose I've set up some axes and a IDLgrPlot that have xranges of [0,8.5]. If
> I want to add a second IDLgrPlot with x data values that go from -1 to 10, and
> have the plot lines end exactly at the axes at x=0 and x=8.5, all I have to do
> is get the cxrange and cyrange from the existing axis, and use this for the
> xrange and yrange of the new IDLgrPlot.
>
> Is this correct?r

---

Subject: Re: plot range in Object Graphics
Posted by davidf on Fri, 12 May 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Brad Gom (b_gom@hotmail.com) writes:

> While I agree that scaling the axes and plot correctly is a pain, and that not scaling
> properly can produce this result, in this case it looks to me like it is a problem with
> version 5.0. The code below, when used in IDL5.0 exhibits the problem, whereas in
> IDL5.3 it looks fine.

Well, now, *that* I can believe. There were LOTS of
problems with IDL 5.0. These objects weren't really
sorted out until about IDL 5.2.

>> P.S. Please don't implement the Polygon idea.
>
> Don't worry. That's why I spend so much time reading this newsgroup.

Actually, as I was informed just a few minutes ago
by my object graphics angel, that wasn't as bad an
idea as I thought it was. But your planned implementation
was a little shaky. I'm hoping I can write an example
soon to show you what I mean.

> Have you ever used duct-tape to patch a radiator hose so that you can get to work?

Oh, many times. Not only on my old beater of a truck,
but in my IDL programs as well. :-)

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: plot range in Object Graphics
Posted by Brad Gom on Fri, 12 May 2000 07:00:00 GMT
View Forum Message <> Reply to Message

David Fanning wrote:

> I really think this is a simple problem of not scaling
> either the data or the axes properly.
> ...
> You have to go get the *actual* values of your axes and
> scale the data to those, or you will get exactly the problems
> you describe.

While I agree that scaling the axes and plot correctly is a pain, and that not scaling properly can produce this result, in this case it looks to me like it is a problem with version 5.0. The code below, when used in IDL5.0 exhibits the problem, whereas in IDL5.3 it looks fine.

**********

```
 thisPlot = Obj_New("IDLgrPLOT", findgen(10), sin(findgen(10)),xrange=[0,8.5],
Color=[255,255,255])
 thisPlot->GetProperty, XRange=xrange, YRange=yrange

 xAxis1 = Obj_New("IDLgrAxis", 0, Color=[255,255,255],Range=xrange, Location=[10, 0
,0], $
    /Exact, name='xaxis1')
 yAxis1 = Obj_New("IDLgrAxis", 1, Color=[255,255,255],Range=yrange, Location=[0,10,0],
$
    /Exact, name='yaxis1')
 yAxis2 = Obj_New("IDLgrAxis", 1, Color=[255,255,255],Range=yrange, Location=[1,10,0],
$
    /Exact, ame='yaxis2')

 xAxis1->GetProperty, CRange=xrange
 yAxis1->GetProperty, CRange=yrange

 xs = Normalize(xrange)
 ys = Normalize(yrange)
```

```
 thisPlot->SetProperty, XCoord_Conv=xs, YCoord_Conv=ys
 xAxis1->SetProperty, XCoord_Conv=xs
 yAxis1->SetProperty, YCoord_Conv=ys
 yAxis2->SetProperty, YCoord_Conv=ys

 thisModel = Obj_New('IDLgrModel')
 thisModel->Add, xAxis1
 thisModel->Add, yAxis1
 thisModel->Add, yAxis2
 thisModel->Add, thisPlot

 thisView = Obj_New('IDLgrView', Viewplane_Rect=[-.2, -.2, 1.3, 1.3], Location=[0,0],
Color=[0,0,0])
 thisView->Add, thisModel
 thisWindow->Draw, thisView

 xAxis1->GetProperty, CRange=xrange
 yAxis1->GetProperty, CRange=yrange
 anotherPlot = Obj_New("IDLgrPLOT", findgen(20)-5.2, sin(findgen(20)),xrange=xrange, $
    yrange=yrange, Color=[255,0,0], XCoord_Conv=xs, YCoord_Conv=ys)
 thisModel->Add, anotherPlot
 thisWindow->Draw, thisView
**************
```

> P.S. Please don't implement the Polygon idea.

Don't worry. That's why I spend so much time reading this newsgroup.

> It's just to grim to imagine and is in the worst tradition of just getting
> the damn thing to work. :-(

Unfortunately, as a physicist who needs to do my own -quick- visualizations while learning IDL at the same time, I have sometimes had to write code that would make a true software engineer shriek in horror in order to meet a deadline. This is, I think, sometimes a necessary evil. A sin, on the other hand, would be never bothering to learn the proper technique, or worse yet, writing improper code that will fall into the hands of other people.

Have you ever used duct-tape to patch a radiator hose so that you can get to work?

Brad