Subject: Re: Plotting Vectors/Rays
Posted by raph on Tue, 07 Oct 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Sorry to follow up my own followup, but I've made progress. A very simple modification (which I'll detail below) to the arrow routine allows for clipping. Does anyone know how I can contact RSI to see if I can get them to fix this?

David Fanning <davidf@dfanning.com> wrote:
> Raph writes:

>> Second, the arrow command seems not to honor the clipping of the plot
>> routine, so it plots arrows in the margins.

> Well, this is a bit trickier. First of all, Arrow is a library routine,
> meaning that it is written in the IDL language. You can find the source
> code in the IDL_DIR/lib subdirectory. Like many library routines, it
> doesn't do *exactly* what you want it to do, so you may have to modify
> it.

This turns out to be even truer that David first thought.

> The problem here is that a vector in IDL direct graphics is described
> as two end points. This means that if the end points extend over
> the plot boundaries that they will not be clipped appropriately,
> even if you use the Clip keyword. (This is essentially what you are
> experiencing with the Arrow command.)

> For example, suppose you typed this command after typing the command
> above:

>  PlotS, [5, 12], [3,-1]

> The line that is drawn extends off the right side of the plot.

> You might try to clip it like this:

>  Erase
>  Plot, Findgen(11), Position=Aspect(1.0)
>  PlotS, [5, 12], [3, -1], Clip=[!X.CRange[0], !Y.CRange[0], $
>  !X.CRange[1], !Y.CRange[1]]

> But even though IDL accepts the command, it doesn't actually
> clip the line. (The PlotS documentation accurately states that
> the PlotS command *accepts* the Clip keyword. It is mute about
> whether is actually *does* anything with the information. :-)

Actually, I found the solution in IDL's online help. According to the description of the CLIP keyword, PLOTS had clipping off by default (why I don't know). As a result in addition to CLIP=[...], one must also specify NOCLIP=0. Not entirely intuitive, but this is IDL. So

```
 PlotS, [5, 12], [3, -1], Clip=[!X.CRange[0], !Y.CRange[0], $
       !X.CRange[1], !Y.CRange[1]], NOCLIP=0
```

does clip correctly.

> In this case, the only way to clip the PlotS line is to make sure
> the endpoints of the line are inside the boundary of the plot.
> For example, you could do it like this:

```
>    Erase
>    Plot, Findgen(11), Position=Aspect(1.0)
>    PlotS, !X.CRange[0] > [5, 12] < !X.CRange[1], $
>          !Y.CRange[0] > [3, -1] < !Y.CRange[1]
```

> Thus, I think the only thing you can do is go into the Arrow
> code and bracket all of the PlotS commands with this
> kind of syntax. (There are not many PlotS commands there.)
> You could bestow the name Arrow_That_Clips_Properly on the
> modified file. :-)

I wandered into this because I couldn't get David's modification to work. Instead I modified ARROW so that it accepts NOCLIP as a keyword and simply passes it to PLOTS. Since this doesn't seem to break anything, why doesn't ARROW include this by default? Or rather how can I get RSI to change make this trivial addition to the standard ARROW.pro?

Raph


---------- ----------
William Raphael Hix Email: raph@phy.ornl.gov
Joint Institute for Heavy Ion Research Voice: (423) 576-8763
Oak Ridge National Laboratory Fax: (423) 576-5780
P.O. Box 2008 (MS 6374)
Oak Ridge, TN 37831