Subject: Re: Plotting Vectors/Rays
Posted by davidf on Fri, 03 Oct 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Raph writes:

> As a new IDL user, I'm just beginning my trek up the learning curve.
> My first trials are plots I know how to do in other plotting packages
> and I'm having trouble getting IDL to do things that are easy in the
> other packages (principally MONGO).
>
> My first trial is to plot vectors (or rays, to avoid using vector in
> a non-IDL fashion) representing nuclear reaction fluxes. Essentially I
> have 4 arrays, X, Y coordinates of the end point, angle and ray length.
>
> Using the plot and arrow commands, I've managed to write a little
> procedure which does the basic ploting, but I have 2 problems.
>
> First, I can't figure out how to make the X and Y axes have the same
> unit length, so that, for example a 45 deg angle is actually 45 degrees
> on the display, not simply a slope of 1.

IDL tries by default to fill up the plotting area, which is probably
what you want most of the time. But certainly not when you are trying
to keep the aspect ratio of the plot constant. If you want a plot of
a specific aspect ratio (e.g. 1.0 in your case), you will have to
calculate and position the plot appropriately in the display window.
Plot positioning is done with the Position keyword to the Plot command.

Fortunately, this work is already done for you in a program called
ASPECT that you can download from my web page. It returns position
coordinates suitable for passing along to the Position keyword of
the Plot command. For example, you can use it like this:

```
   Plot, Findgen(11), Position=Aspect(1.0)
```

> 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.

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. :-)

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. :-)

Cheers,

David

David Fanning, Ph.D.
Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438
Coyote's Guide to IDL Programming: http://www.dfanning.com/