Subject: Re: Plotting Vectors/Rays
Posted by raph on Mon, 06 Oct 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Hi-

I think I need to follow-up. First and foremost, thanks for all the help.
However the advice given doesn't completely solve my problems, as I'll explain below.

David Fanning wrote:
>
> Raph writes:
>
>> 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)

Setting Aspect to 1.0 seems to produces a physically square plot. What I want is a plot with physically orthonormal axes. If my x and y ranges are equal, then aspect =1.0 does this job. But for un-equal axis ranges, I think I need to set the aspect equal to the ratio of the x and y axis lengths. But this becomes a chicken and egg problem, as I need the axis ranges to call Plot, which determines the axis ranges.

Now, it is my intention to specify the range of the Y-axis, but that still leaves the x-axis. I guess I could write a procedure to search the data set
finding the min and max x for a given range of y. Is there a pre-existing routine to do this?

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

>
> 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 tried this, modifying line 112 of a copy of arrow.pro to look like

```
   plots, !X.CRange[0] > [xp0, xp1] < !X.CRange[1], $
          !Y.CRange[0] > [yp0, yp1] < !Y.CRange[1], $
          /DEVICE, COLOR = color, THICK = thick
```

But IDL doesn't seem to like this saying

```
 IDL> .run arrow_clip

          plots, !X.CRange[0] > [xp0, xp1] < !X.CRange[1], $
                      ^
% Syntax error.
  At: arrow_clip.pro, Line 112
% 1 Compilation errors in module ARROW_CLIP.
```

What am I doing wrong?

 ---------- ----------
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