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/

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

In article <34392838.41C6@phy.ornl.gov>, William Raphael Hix
<raph@phy.ornl.gov> wrote:

> 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.
>> . . .

I'm sure there is a better way to do this but here is my routine;
unfortunately it creates a blank plot first to get the right data ranges.
It would really be useful to have a standard /isotropic keyword like for
map_set. I haven't tested it with IDL 5 yet.

```
pro even_scale, x, y
; CREATE EVEN (exact) SCALE frame with equal data units plotting as equal
;       distances on the X and Y axes
; This is extracted from  Plot_even.pro  created by
; Bobby Candey, Atlantic Research Corp.   1991 April 11; S. F. Fung 9/10/91.
; now Robert.M.Candey@gsfc.nasa.gov
; alters !p.position and !p.multi and !x/y.range and makes a dummy plot
!p.position = 0
!p.multi = 0 ; reset to 1 plot per page
if (n_params(0) eq 2) then begin
  !x.range = [min(x), max(x)]
  !y.range = [min(y), max(y)]
endif
plot,[0,1],[0,1] ; dummy plot to set !x/y.crange and !x/y.window
device,/close
wx = !x.window & wy = !y.window ; window size in normalized coordinates
xr = !x.crange(1) - !x.crange(0) ; data range in data coordinates
yr = !y.crange(1) - !y.crange(0)
xdevice = wx * float(!d.x_size) ; current window in pixels
ydevice = wy * float(!d.y_size)
xrdevice = xdevice(1) - xdevice(0) ; current window width in pixels
```

```
yrdevice = ydevice(1) - ydevice(0)
xratio = xr / xrdevice ; data units per pixel
yratio = yr / yrdevice
ratio = max([xratio, yratio])
newxmaxnorm = (xr / ratio + xdevice(0)) / float(!d.x_size)
newymaxnorm = (yr / ratio + ydevice(0)) / float(!d.y_size)
!p.position = [wx(0), wy(0), newxmaxnorm, newymaxnorm]
return
end
```

--
Robert.M.Candey@gsfc.nasa.gov
NASA Goddard Space Flight Center, Code 632
Greenbelt, MD 20771 USA   1-301-286-6707

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

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

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

William Raphael Hix <raph@phy.ornl.gov> writes:

> 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.
>>

You might try my routine SETSCALE which is available from

ftp://sohoftp.nascom.nasa.gov/solarsoft/gen/idl/display/

This should do what you want.

Bill Thompson