
Subject: Re: Plotting a compass

Posted by [David Fanning](#) on Fri, 09 Nov 2012 19:36:01 GMT

[View Forum Message](#) <> [Reply to Message](#)

Davide writes:

> Hi guys, I am trying to plot something like a compass on top of an image (a 2D matrix filled with floats. It is supposed to be a flux map for an astronomical object. No reliable wcs system inside). So I ended up using "arrow".
> I am having difficulties plotting two orthogonal arrows. The code (see below) seems fine to me. Is that something related to some rescaling? (Or maybe I made some awkward mistake).
> Also, can you suggest something more up to date than tvscale? Consider that my goal is have maps with a color bar and spatial scales along x and y.

Oh, dear! :-(

I don't know where to start. I guess I'd start by updating your Coyote Library to something that was written in the, I don't know, last 10 years or so. Things have changed. In fact, things have changed TODAY!

<http://www.idlcoyote.com/programs/coyoteprograms.zip>

Humm. The cgImage program is perfectly capable of adding axes with different ranges so you can set up a data coordinate system, etc. And cgArrow can make prettier arrows than you are making.

I guess if I were going to think about drawing orthogonal vectors I would be thinking about working in polar coordinates, rather than rectangular coordinates. The angle thing would be a LOT easier! You can use CV_COORD to do the conversions for you.

Weird dimensions on the PostScript file, too. What are you going to do with that thing when you are done with it?

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: Plotting a compass

Posted by [DAVIDE LENA](#) on Fri, 09 Nov 2012 22:26:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

Thanks.

So, how do I put a colorbar when I use `cgImage` to print on a `eps`?
`cgColorbar` comes after the instruction that directs the output to the `eps`.

```
cgIMAGE, f, POSITION=p, /KEEP_ASPECT_RATIO, MINVALUE = 0,  
outfile='wonderfulplot.eps', OUTPUT = 'EPS'
```

```
cgColorbar, FORMAT='(F2.0)', Position=[p[2], p[1], p[2]+0.015, p[3] ], ncolors=256, /vertical,  
/right, Divisions=4, Range=[minc, maxc]
```

For what concerns the arrows, I guess that the problem is that I am using normal coordinates, that's why they are not orthogonal. I should switch to the data coordinates. Anyway, they look the same than the arrows obtained with the old function.

On Friday, November 9, 2012 2:36:01 PM UTC-5, David Fanning wrote:

> Davide writes:

>

>

>

>> Hi guys, I am trying to plot something like a compass on top of an image (a 2D matrix filled with floats. It is supposed to be a flux map for an astronomical object. No reliable wcs system inside). So I ended up using "arrow".

>

>> I am having difficulties plotting two orthogonal arrows. The code (see below) seems fine to me. Is that something related to some rescaling? (Or maybe I made some awkward mistake).

>

>> Also, can you suggest something more up to date than `tvscale`? Consider that my goal is have maps with a color bar and spatial scales along x and y.

>

>

>

> Oh, dear! :-(

>

>

>

> I don't know where to start. I guess I'd start

>

> by updating your Coyote Library to something

>

> that was written in the, I don't know, last 10
>
> years or so. Things have changed. In fact, things
>
> have changed TODAY!
>
>
>
> <http://www.idlcoyote.com/programs/coyoteprograms.zip>
>
>
>
> Humm. The cgImage program is perfectly capable of adding
>
> axes with different ranges so you can set up a data
>
> coordinate system, etc. And cgArrow can make prettier
>
> arrows than you are making.
>
>
>
> I guess if I were going to think about drawing orthogonal
>
> vectors I would be thinking about working in polar coordinates,
>
> rather than rectangular coordinates. The angle thing would
>
> be a LOT easier! You can use CV_COORD to do the conversions
>
> for you.
>
>
>
> Weird dimensions on the PostScript file, too. What are
>
> you going to do with that thing when you are done with it?
>
>
>
> Cheers,
>
>
>
> David
>
>
>

>
>
>
>
>
> --
>
> David Fanning, Ph.D.
>
> Fanning Software Consulting, Inc.
>
> Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
>
> Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: Plotting a compass
Posted by [David Fanning](#) on Fri, 09 Nov 2012 22:47:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

Davide writes:

> So, how do I put a colorbar when I use cgImage to print on a eps?
> cgColorbar comes after the instruction that directs the output to the eps.
>
> cgIMAGE, f, POSITION=p, /KEEP_ASPECT_RATIO, MINVALUE = 0,
outfilename='wonderfulplot.eps', OUTPUT = 'EPS'
> cgColorbar, FORMAT='(F2.0)', Position=[p[2], p[1], p[2]+0.015, p[3]], ncolors=256, /vertical,
/right, Divisions=4, Range=[minc, maxc]
>

I would do it pretty much the way you were doing it before:

```
PS_Start, 'wonderfulplot.eps'  
cgIMAGE, f, POSITION=p, /KEEP_ASPECT_RATIO, MINVALUE = 0  
cgColorbar, FORMAT='(F2.0)', Position=[p[2],p[1],p[2]+0.015 p[3]], $  
    ncolors=256, /vertical, /right, Divisions=4, Range=[minc, maxc]  
PS_End
```

> For what concerns the arrows, I guess that the problem is that I am using normal coordinates, that's why they are not orthogonal. I should switch to the data coordinates. Anyway, they look the same than the arrows obtained with the old function.

Well, you will certainly have to take into account the aspect ratio of the plotting window if you are going to use normalized coordinates. But, that is just another good reason to use polar coordinates. R and Theta are both going to be easy to calculate! :-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: Plotting a compass

Posted by [Dick Jackson](#) on Sat, 10 Nov 2012 02:01:18 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Davide,

Just to add a data point, there's nothing wrong with how you are computing the xy points, getting the perpendicular slope by taking $-1/m1$.

```
x0 = 0.8
y0 = 0.75
x1 = 0.85
y1 = 0.77
```

```
; slope of the first arrow:
m1 = (y1 - y0)/(x1 - x0)
```

```
x2 = 0.78
y2 = y0 - 1./m1 * (x2 - x0)
```

```
Plot,[x0,x1],[y0,y1],/Isotropic,xr=[0,1],yr=[0,1]
oPlot,[x0,x2],[y0,y2]
```

Using data coordinates with isotropic axes shows nice perpendicular line segments. Using normalized coordinates with a non-square drawing area will not! :-) You'll need to scale x or y lengths by the aspect ratio. To me that seems easier than using polar coords, but I may be missing something. Hope this helps!

Cheers,
-Dick

Dick Jackson Software Consulting
Victoria, BC, Canada

On Friday, November 9, 2012 2:47:48 PM UTC-8, David Fanning wrote:

> Davide writes:

>

>

>

>> So, how do I put a colorbar when I use `cglImage` to print on a eps?

```

>
>> cgColorbar comes after the instruction that directs the output to the eps.
>
>>
>
>> cgIMAGE, f, POSITION=p, /KEEP_ASPECT_RATIO, MINVALUE = 0,
outfilename='wonderfulplot.eps', OUTPUT = 'EPS'
>
>>   cgColorbar, FORMAT='(F2.0)', Position=[p[2], p[1], p[2]+0.015, p[3] ], ncolors=256, /vertical,
/right, Divisions=4, Range=[minc, maxc]
>
>>
>
> I would do it pretty much the way you were doing it before:
>
>
>
> PS_Start, 'wonderfulplot.eps'
>
> cgIMAGE, f, POSITION=p, /KEEP_ASPECT_RATIO, MINVALUE = 0
>
> cgColorbar, FORMAT='(F2.0)', Position=[p[2],p[1],p[2]+0.015 p[3]], $
>
>   ncolors=256, /vertical, /right, Divisions=4, Range=[minc, maxc]
>
> PS_End
>
>
>
>> For what concerns the arrows, I guess that the problem is that I am using normal coordinates,
that's why they are not orthogonal. I should switch to the data coordinates. Anyway, they look the
same than the arrows obtained with the old function.
>
>
>
>
> Well, you will certainly have to take into account the aspect
>
> ratio of the plotting window if you are going to use normalized
>
> coordinates. But, that is just another good reason to use polar
>
> coordinates. R and Theta are both going to be easy to calculate! :-)
>
>
>
> Cheers,
>
>

```

>
> David
>
> --
>
> David Fanning, Ph.D.
>
> Fanning Software Consulting, Inc.
>
> Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
>
> Sepore ma de ni thue. ("Perhaps thou speakest truth.")
