
Subject: Re: Adding extra white in middle of colour bar
Posted by [Sir Loin Steak](#) on Thu, 22 Aug 2013 12:57:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Thursday, 22 August 2013 01:13:41 UTC+1, David Fanning wrote:

> ljs15@fsmail.net writes:

>

>

>

>> But this does bring me on to something else I've been wondering about regarding contours and fill colours (sorry for all the questions!).

>

>>

>

>> Say I had data and wanted to contour it from -10 to +10 using a blue-white-red colour scale. I would normally do:

>

>>

>

>> ncont = 21

>

>> cgloadct, 22, /brewer, /reverse, ncolors=ncont, /silent

>

>> clev = scale_vector(findgen(ncont), -maxval, maxval)

>

>> ccol = bindgen(ncont)

>

>>

>

>> Then contour it using levels=clev, c_colors=ccol.

>

>>

>

>> However, I actually want [-10,-9] to be one colour, [-9,-8] to be another colour etc. So I actually need 20 fill colours not 21.

>

>>

>

>> If I select ncont=20 then my levels do not come out as integer values, but using ncont=21 means the colour bar is not divided neatly in two, with red colours > 0 and blue colours < 0.

>

>>

>

>> Am I misunderstanding how IDL fills in the contours? Is there a way around this problem? I assume that if I set the levels to [-1,0,1] and tell IDL to fill the contour plot, then it fills [-1,0],[0,1],[1,1+].

>

>

```

>
> Well, you can test this easily enough.
>
>
>
> data = cgScaleVector(Dist(51), -10, 10)
>
> pos = [0.125, 0.125, 0.925, 0.8]
>
> cgColorFill, Position=pos, Color='charcoal'
>
> cgLoadCT, 22, /Brewer, /Reverse, NColors=3, Bottom=1
>
> cgContour, data, Levels=[-1, 0, 1], C_Colors=Bindgen(3)+1B, $
>   Position=pos, /NoErase, /Outline, /Fill
>
>
>
> So, you are exactly right. You see from this plot that anything less
> than -1 was filled with the background color (charcoal) and that
> anything greater than 1 was filled with the red color.
>
>
>
> The color bar for such a plot would look like this:
>
>
>
> cgcolorbar, ncolors = 2, oob_low='charcoal', oob_high=3, $
>   range=[-1,1], bottom=1, xticks=3, xminor=0
>
>
>
> If you are going to use out-of-bounds colors, then the number of colors
> in your color bar is always going to be one less than the number of
> levels in your contour plot.
>
>
>
> If you are NOT going to use out-of-bounds colors (this is what I think
> you are asking about), then the number of contour levels you need is

```

```

>
> always one more than the number of colors you want to use. Think of how
>
> many vertical lines you have to draw in a color bar to separate or
>
> block-off X number of colors. For three colors, for example, you have to
>
> draw a vertical line on the left, where the color starts, one between
>
> the first and second color, one between the second and third color, and
>
> one to show where the third color ends.
>
>
>
> Consider this:
>
>
>
>   cgLoadCT, 0
>
>   TVLCT, cgColor(['blu6','grn6','red6'], /Triple), 1
>
>   levels = [-1,0,1,2]
>
>   cgContour, data, Levels=levels, C_Colors=Bindgen(3)+1B, $
>
>     Position=pos, /NoErase, /Outline, /Fill
>
>
>
> Yikes! We have blue, green, and red contours, but why is everything
>
> greater than 2 the same color as the -1 to 0 level? Well, because we
>
> have the "greater than 2" contour to draw, and the colors recycle. In
>
> this case, the fourth color is the same as the first, the fifth the same
>
> as the second, etc. So, to get what we want we have to load another
>
> color and tell the contour plot to load it. Let's make the fourth color
>
> the same as the background color, like this:
>
>
>
>   cgLoadCT, 0

```

```
>
> TVLCT, cgColor(['blu6','grn6','red6'], /Triple), 1
>
> TVLCT, cgColor('white', /Triple), 4
>
> levels = [-1,0,1,2]
>
> cgContour, data, Levels=levels, C_Colors=Bindgen(4)+1B, $
>
>     Position=pos, /NoErase, /Outline, /Fill
>
>
>
> But, now, what about the color bar? It uses three colors, and the range
>
> is -1 to 2.
>
>
>
> cgcolorbar, ncolors = 3, range=[-1,2], bottom=1, xticks=3, xminor=0
>
>
>
> In other words, it's pretty damm confusing. :-)
>
>
>
> Cheers,
>
>
>
> David
>
>
>
> --
>
> David Fanning, Ph.D.
>
> Fanning Software Consulting, Inc.
>
> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
>
> Sepore ma de ni thue. ("Perhaps thou speakest truth.")
```

Brilliant! It actually makes sense now.

Thanks very much for that detailed reply. Sometimes I wonder how the IDL community would do without you!
