
Subject: Re: Logarithmic Color Scaling
Posted by [Paolo Grigis](#) on Tue, 05 Dec 2006 16:07:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

Is this the result you would like to get? It looks ugly enough though...

<http://www.astro.phys.ethz.ch/staff/pgrigis/coltroubles.png>

I generated this using spectro_plot from solarsoft:

```
x=findgen(10000)
data=rebin(x,10000,10)
y=findgen(10)
```

```
spectro_plot,data,x,y,/no_ut,/xlog,xrange=[1,1d4],ystyle=1,position=[0.1,0.7,0.9,0.8],/noerase
```

Ciao,
Paolo

David Fanning wrote:

```
> Folks,
>
> I'm embarrassed to admit this, but I spent the entire day
> yesterday working on a logarithmic color scaling problem
> and got absolutely nowhere. I was really counting on a
> breakthrough in the shower this morning, but no joy there,
> either. :-(
>
> My dilemma is this. I can produce a log scaled image
> (using LOGSCL) and I can create a log scaled color table
> (again using LOGSCL with the method Lagos outlined yesterday).
> What I cannot do is associate a color on the color bar
> with the actual image value.
>
> In other words, when I click on the image, I can read that
> value back from the image. The value of that pixel does
> not correspond to the color representing that value
> in the color table. In fact, it is not even close. It is
> so far off, in fact, that it makes me think there is something
> absolutely fundamental that I am not understanding about the
> problem.
>
```

> I have thought about nothing else for 24 hours and can't
> see my way out of this problem. (Although a bulky furnace
> is going to demand some of my time today.) Has anyone
> EVER done this successfully? Could you show me a bit of
> code?
>
> I want to show a data set with values extending over
> several decades with a logarithmic color bar. The
> data set I am using is this one:
>
> image = FltArr(400, 400)
> image[30:40, 30:40] = 10
> image[50:60, 50:60] = 100
> image[70:80, 70:80] = 1000
> image[90:100, 90:100] = 2500
> image[110:120, 110:120] = 3500
> image[130:140, 130:140] = 5000
> image[150:160, 150:160] = 7500
> image[170:180, 170:180] = 10000
>
>
> I can show this data set logarithmically scaled:
>
> LoadCT, 33
> TV, LogScl(image)
>
> And I can even show the logarithmically scaled
> color values that accurately reflects the image
> values:
>
> TVLCT, r, g, b, /GET
> TVLCT, r[LogScl(index)], g[LogScl(index)], b[LogScl(index)]
> Colorbar, range=[1,10000]
>
> What I cannot show is a Colorbar with a logarithmic axis scale
> that accurately shows the image colors. :-(
>
> Colorbar, Range=[1,10000], XLOG=1, XTICKS=0, MINOR=5
>
> I am VERY open to ideas. :-)
>
> Cheers,
>
> David
>

Subject: Re: Logarithmic Color Scaling
Posted by [David Fanning](#) on Tue, 05 Dec 2006 16:15:26 GMT
[View Forum Message](#) <> [Reply to Message](#)

Paolo Grigis writes:

```
> Is this the result you would like to get? It looks ugly
> enough though...
>
> http://www.astro.phys.ethz.ch/staff/pgrigis/coltroubles.png
>
> I generated this using spectro_plot from solarsoft:
>
> x=findgen(10000)
> data=rebin(x,10000,10)
> y=findgen(10)
>
> spectro_plot,data,x,y,/no_ut,/xlog,xrange=[1,1d4],ystyle=1,position=[0.1,0.7,0.9,0.8],/noerase
```

I guess that's what I am looking for, yes.
I looked into the SOLARSOFT software yesterday, but
I didn't think I had the 3 terra bytes of data space
on my computer to hold it. :-)

Uh, do you think you could e-mail the relevant part of
the spectro_plot code? I just want to see how it is done. :-)

Cheers,

David

--

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

Subject: Re: Logarithmic Color Scaling
Posted by [Paolo Grigis](#) on Tue, 05 Dec 2006 16:32:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

Actually, you only need the /gen/idl part of
solarsoft, which luckily is only 19 MB of data.

All this routines depends strongly on each other,
so you might have some digging to do to find out
how this works...

You can find it on the ftp server

sohoftp.nascom.nasa.gov

in the subdir

solarsoft/gen/idl

If you have an ftp client which can download the content of directories recursively, it should be no trouble to get it.

"add_path.pro" (located in /system) makes it easy to update your path to include the tree.

Ciao,
Paolo

David Fanning wrote:

> Paolo Grigis writes:

>

>

>> Is this the result you would like to get? It looks ugly

>> enough though...

>>

>> <http://www.astro.phys.ethz.ch/staff/pgrigis/coltroubles.png>

>>

>> I generated this using spectro_plot from solarsoft:

>>

>> x=findgen(10000)

>> data=rebin(x,10000,10)

>> y=findgen(10)

>>

>> spectro_plot,data,x,y,/no_ut,/xlog,xrange=[1,1d4],ystyle=1,position=[0.1,0.7,0.9,0.8],/noerase

>

>

> I guess that's what I am looking for, yes.

> I looked into the SOLARSOFT software yesterday, but

> I didn't think I had the 3 terra bytes of data space

> on my computer to hold it. :-(

>

> Uh, do you think you could e-mail the relevant part of

> the spectro_plot code? I just want to see how it is done. :-)

>

> Cheers,

>

> David

Subject: Re: Logarithmic Color Scaling

Posted by [David Fanning](#) on Tue, 05 Dec 2006 16:35:54 GMT

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

Paolo Grigis writes:

- > Actually, you only need the /gen/idl part of
- > solarsoft, which luckily is only 19 MB of data.
- >
- > All this routines depends strongly on each other,
- > so you might have some digging to do to find out
- > how this works...

Why does this remind me of iTools... :-(

I'll give it a try. Thanks.

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

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

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

Subject: Re: Logarithmic Color Scaling

Posted by [JD Smith](#) on Tue, 05 Dec 2006 18:16:41 GMT

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

On Tue, 05 Dec 2006 08:16:14 -0700, David Fanning wrote:

- > Folks,
- >
- > I'm embarrassed to admit this, but I spent the entire day yesterday
- > working on a logarithmic color scaling problem and got absolutely nowhere.
- > I was really counting on a breakthrough in the shower this morning, but no
- > joy there, either. :-(
- >
- > My dilemma is this. I can produce a log scaled image (using LOGSCL) and I
- > can create a log scaled color table (again using LOGSCL with the method
- > Lagos outlined yesterday). What I cannot do is associate a color on the
- > color bar with the actual image value.

My personal opinion is to keep the scaling of the image data, and the mapping of image data over some min->max range to colors on the display as separate. The former can be quite flexible, log, sqrt, asinh, whatever. The latter should be linear, and reflect the mapping using axes which properly map original data values to colors. Why do

I make the division this way? Presumably the data are floating point or double floats, and can take much more extreme scaling before they begin to suffer from roundoff and other numerical concerns. Not so with a 256 element byte color table.

In that context, I think you are double-logging. I.e. you are scaling your data logarithmically, and then separately scaling your color map *and* the colorbar axis as well. This could explain why your values don't match up.

You could either a) just display the linear color-bar (i.e. what you actually used, with logarithmic axes of course), or b) load a logarithmically mapped color bar as you do first thing, and run the *linear* image data through it and display with a linear axis, or c) use a linear color table with a log scaled image, display this log-scaled color bar, but then use a linear X axis values. You can't both map the colors *and* map the axis values, that's "double-logging".

There are four places log could get applied, two each for data and colorbar. You must pick one on each side of the equation.

- 1) to the data themselves
- 2a) to the colormap indices (for displaying data)
- 2b) to the colormap indices (for displaying colorbar)
- 3) to the axis of the colorbar

Here are the possibilities (3 of which I discussed above):

- a: 1 (data side) + 3 (colorbar side)
- b: 2a (data side) + 2b (colorbar side)
- c: 1 (data side) + 2b (colorbar side)
- d: 2a (data side) + 3 (colorbar side)

you are now using:

- e: 1 (data side) + 2b (colorbar side) + 3 (colorbar side)

My preference, which keeps things simple, is a). This would be especially true if you implemented one of the scaling functions used in the Spitzer community: LogLog. That could get confusing fast ;).

One wrinkle is if you don't use a colorbar axis. Then a) doesn't work so well. In that case, you can use c), with an "implied" linear x-axis.

JD

Subject: Re: Logarithmic Color Scaling
Posted by [Braedley](#) on Tue, 05 Dec 2006 20:06:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

JD Smith wrote:

> On Tue, 05 Dec 2006 08:16:14 -0700, David Fanning wrote:
>
>> Folks,
>>
>> I'm embarrassed to admit this, but I spent the entire day yesterday
>> working on a logarithmic color scaling problem and got absolutely nowhere.
>> I was really counting on a breakthrough in the shower this morning, but no
>> joy there, either. :-(
>>
>> My dilemma is this. I can produce a log scaled image (using LOGSCL) and I
>> can create a log scaled color table (again using LOGSCL with the method
>> Lagos outlined yesterday). What I cannot do is associate a color on the
>> color bar with the actual image value.
>
> My personal opinion is to keep the scaling of the image data, and the
> mapping of image data over some min->max range to colors on the
> display as separate. The former can be quite flexible, log, sqrt,
> asinh, whatever. The latter should be linear, and reflect the mapping
> using axes which properly map original data values to colors. Why do
> I make the division this way? Presumably the data are floating point
> or double floats, and can take much more extreme scaling before they
> begin to suffer from roundoff and other numerical concerns. Not so
> with a 256 element byte color table.
>
> In that context, I think you are double-logging. I.e. you are scaling
> your data logarithmically, and then separately scaling your color map
> *and* the colorbar axis as well. This could explain why your values don't
> match up.
>
> You could either a) just display the linear color-bar (i.e. what you
> actually used, with logarithmic axes of course), or b) load a
> logarithmically mapped color bar as you do first thing, and run the
> *linear* image data through it and display with a linear axis, or c)
> use a linear color table with a log scaled image, display this
> log-scaled color bar, but then use a linear X axis values. You can't
> both map the colors *and* map the axis values, that's "double-logging".
>
> There are four places log could get applied, two each for data and
> colorbar. You must pick one on each side of the equation.
>
> 1) to the data themselves
> 2a) to the colormap indices (for displaying data)
> 2b) to the colormap indices (for displaying colorbar)
> 3) to the axis of the colorbar

>
> Here are the possibilities (3 of which I discussed above):
>
> a: 1 (data side) + 3 (colorbar side)
> b: 2a (data side) + 2b (colorbar side)
> c: 1 (data side) + 2b (colorbar side)
> d: 2a (data side) + 3 (colorbar side)
>
> you are now using:
>
> e: 1 (data side) + 2b (colorbar side) + 3 (colorbar side)
>
> My preference, which keeps things simple, is a). This would be
> especially true if you implemented one of the scaling functions used
> in the Spitzer community: LogLog. That could get confusing fast ;).
>
> One wrinkle is if you don't use a colorbar axis. Then a) doesn't work so
> well. In that case, you can use c), with an "implied" linear x-axis.
>
> JD

I would tend to agree with JD on this. I don't think it would be a good idea to scale the color table (and hence the colorbar). I've never had to worry about the problem you're facing because whenever I did anything remotely like this, I worked in dB, and so my colorbar was labeled in dB. I think it would be trivial for you to go from this thinking to labeling your colorbar logarithmically.

Braedley

Subject: Re: Logarithmic Color Scaling
Posted by [David Fanning](#) on Tue, 05 Dec 2006 20:20:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

Braedley writes:

> I would tend to agree with JD on this. I don't think it would be a
> good idea to scale the color table (and hence the colorbar). I've
> never had to worry about the problem you're facing because whenever I
> did anything remotely like this, I worked in dB, and so my colorbar was
> labeled in dB. I think it would be trivial for you to go from this
> thinking to labeling your colorbar logarithmically.

Well, I've been looking for the trivial solution for the past 48 hours straight. I'd be delighted to be pointed in the right direction. May I see your code?

I do agree with you that it is trivial to label a color bar logarithmically. What I am finding difficult is making the colors associated with the color bar labels accurately reflect the values in the data, which seems to me to be pretty much the point of a color bar.

With the exception of `spectro_plot` from the SolarSoft library, which I have still not had time to download, I have seen a lot of advice, but no working code. This leads me to believe maybe it's not as trivial as everyone (including me) thought it might be. :-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

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

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

Subject: Re: Logarithmic Color Scaling

Posted by [JD Smith](#) on Tue, 05 Dec 2006 21:21:47 GMT

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

On Tue, 05 Dec 2006 13:20:11 -0700, David Fanning wrote:

> Braedley writes:

>

>> [quoted text muted]

>

> Well, I've been looking for the trivial solution for
> the past 48 hours straight. I'd be delighted to be
> pointed in the right direction. May I see your code?

>

> I do agree with you that it is trivial to
> label a color bar logarithmically. What I am finding
> difficult is making the colors associated with the
> color bar labels accurately reflect the values
> in the data, which seems to me to be pretty much the
> point of a color bar.

>

> With the exception of `spectro_plot` from the SolarSoft
> library, which I have still not had time to download,
> I have seen a lot of advice, but no working code. This

> leads me to believe maybe it's not as trivial as everyone
> (including me) thought it might be. :-)

I'm not sure what the big mystery is. Here's a simple code illustrating my case 'a' (the "don't scale the colorbar indices" case):

```
loadct,0
d=dist(256) + 1.

;; Set the color where d=100 to red
index100=round(alog10(100.)/(alog10(max(d))-alog10(min(d)))* 256)
tv!ct,255,0,0,index100

tvsc!,alog10(d),10,10
colorbar=rebin(indgen(256),256,60)
tv,colorbar,10,(yb=10+256+20)
plot,[0],[0],/NODATA,/NOERASE,POSITION=[10,yb,256+10,yb+60], $
    /DEVICE,XRANGE=[min(d),max(d)],/XLOG,/XSTYLE,YSTYLE=4,XTICKL EN=.15, $
    COLOR=2*!D.TABLE_SIZE/3
```

JD

Subject: Re: Logarithmic Color Scaling
Posted by [Braedley](#) on Wed, 06 Dec 2006 13:18:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

> Braedley writes:
>
>> I would tend to agree with JD on this. I don't think it would be a
>> good idea to scale the color table (and hence the colorbar). I've
>> never had to worry about the problem you're facing because whenever I
>> did anything remotely like this, I worked in dB, and so my colorbar was
>> labeled in dB. I think it would be trivial for you to go from this
>> thinking to labeling your colorbar logarithmically.
>
> Well, I've been looking for the trivial solution for
> the past 48 hours straight. I'd be delighted to be
> pointed in the right direction. May I see your code?
>
> I do agree with you that it is trivial to
> label a color bar logarithmically. What I am finding
> difficult is making the colors associated with the
> color bar labels accurately reflect the values
> in the data, which seems to me to be pretty much the

> point of a color bar.
>
> With the exception of spectro_plot from the SolarSoft
> library, which I have still not had time to download,
> I have seen a lot of advice, but no working code. This
> leads me to believe maybe it's not as trivial as everyone
> (including me) thought it might be. :-)
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Actually, I got the code I'm using just for the colorbar from a co-worker, and may very well have been written by you back when you were at Precision Visuals. The trick is when the data is byte scaled, the extremes of the data need to be recorded. Since the data I use could extend from 0dB down to well under -120dB, the decision was made to impose a maximum dynamic range of 80dB for the plot. The data is then byte scaled like so:

```
max=max(data)
new_data=bytsc1(data, max=max, min=max-80)
;get the data on the screen using combination of plot, /nodata and
tvimage
```

Remember that the data is in dB. Then the call to a colorbar routine just needs a log keyword, and a call like so:

```
some_colorbar_routine, /top, min=10.0^((max-80)/10.0),
max=10^(max/10.0), /log
```

Braedley
