
Subject: Re: isurface with custom palette
Posted by [penteado](#) on Tue, 12 Jan 2010 17:05:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Jan 12, 2:34 pm, gg <qonq...@googlemail.com> wrote:
> On Jan 12, 5:23 pm, pp <pp.pente...@gmail.com> wrote:
>
>> On Jan 12, 2:12 pm, gg <qonq...@googlemail.com> wrote:
>
>>> Thanks a lot! This is exactly what I needed. Now, if I am not asking
>>> too much, I would like to know if is posible to insert colorbar with
>>> data_colors values but not as specified by the RGB_TABLE keyword?
>
>> I am not sure of what you mean. Do you want the colorbar labels to be
>> values like
>> [-1000,1,25,50,100,150,250,350,500,750,1000,1250,1500,1750,2 000,2500,3000]
>> corresponding to the colors you defined in rgb_table? Or is it other
>> values?
>
> Yes, I would like to have colorbar values exactly with the values
> [-1000,1,25,50,100,150,250,350,500,750,1000,1250,1500,1750,2 000,2500,3000]

Then, it would be just a matter of adding /insert_colorbar to the
isurface call.

But my guess is you will not like the result. The reason is the way
you created your colortable. It has 256 levels, where the first 18 are
those discrete colors you set, and the next 238 you do not use in your
surface. But the colorbar inserted will have all the 256 colors, not
just the 18 you used, thus the extra 238 odd colors at its end.

One way to solve this is to change the definition of the colorbar, so
that it spans uniformly the 0-255 range, with 18 regions of repeated
values, since you want only 18 discrete levels. It can be done with
congrid to expand the 0-17 0-255 ranges with nearest neighbour
interpolation (thus keeping the levels discrete):

```
rgb_e=congrid(rgb_table[0:17,*],256,3)
levels_e=congrid(levels,256)
data_colors=value_locate(levels_e,data)
isurface,data,rgb_table=rgb_e,vert_colors=data_colors,/insert_colorbar
```
