Subject: Box Axes with Map Function
Posted by David Fanning on Sun, 02 Dec 2012 16:53:58 GMT
View Forum Message <> Reply to Message

Folks,

I *finally* discovered a way to create box axes with the function graphics Map function that are labeled in the way we expect box axes to be labeled. There are two keys.

First, make sure you use a LIMIT keyword on your map projection and make sure the LATITUDE limits do not touch -90 or 90 degrees. Without this, you will get extraneous labels on your map.

Second, use two map projections so you can label the axes appropriately.

Here is the code.

```
  mp1 = map('Equirectangular', CENTER_LONGITUDE=180, $
    POSITION=[0.1,0.1,0.90,0.75], $
    LABEL_POSITION = 0, BOX_AXES=1, $
    GRID_LATITUDE = 30, GRID_LONGITUDE = 45, $
    /CURRENT, ASPECT_RATIO=0, LIMIT=[-89.99, 0, 89.99, 360])
  mp1['Latitudes'].label_angle=90
  mp1['Longitudes'].label_angle=0

  ; A second map projection is required to create the
  ; box axes correctly.
  mp2 = map('Equirectangular', CENTER_LONGITUDE=180, $
    POSITION=[0.1,0.1,0.90,0.75], $
    LABEL_POSITION = 1, BOX_AXES=1, $
    GRID_LATITUDE = 30, GRID_LONGITUDE = 45, $
    /CURRENT, ASPECT_RATIO=0, LIMIT=[-89.99, 0, 89.99, 360])
  mp2['Latitudes'].label_angle=270
  mp2['Longitudes'].label_angle=0

  ; Something to see on the map.
  c = MapContinents(Color=cgColor('tomato', /Triple, /Row))
```

I believe this requires IDL 8.2.1 to work correctly, although I am not certain of this.

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: Box Axes with Map Function
Posted by David Fanning on Tue, 04 Dec 2012 15:36:55 GMT
View Forum Message <> Reply to Message

Paul van Delst writes:

> If it makes you feel any better, I don't see how Alain's answer is any
> more "correct" compared to yours.

You are probably right. Initially, I thought it would be faster than mine, because there is a time penalty in creating two map projections. But, as it happened, mucking around with each label of the grid object with a FOREACH loop was even slower.

Still, I think it is a more "elegant" solution. Creating two of everything, just so you can get things to "work" smacks of hacksterism to me. :-)

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: Box Axes with Map Function
Posted by Paul Van Delst[1] on Tue, 04 Dec 2012 17:05:32 GMT
View Forum Message <> Reply to Message

Hello,

On 12/04/12 10:36, David Fanning wrote:
> Still, I think it is a more "elegant" solution. Creating

> two of everything, just so you can get things to "work"
> smacks of hacksterism to me. :-)

Oh, undoubtedly. But that problem has to be addressed by Execlis.
(Excelis? Ecxelis? I can never remember).

I'm actually on the verge of deciding whether to plunge into all this
with respect to analysing satellite data simulations. I have to say
that, given all the Function Graphic MAP related posts over the last
couple months, GrADS[*] is looking pretty convenient right now.

But I kid - your cg* system would be used first. :o)

And, while it may be little consolation, don't underestimate the value
of your articles exposing the tricks to get IDL FG Maps working "right".
There are orders of magnitude more lurkers checking out those pages than
corresponders.

cheers,

paulv

[*] http://www.iges.org/grads/

Subject: Re: Box Axes with Map Function
Posted by Fabzi on Tue, 04 Dec 2012 17:30:38 GMT
View Forum Message <> Reply to Message

On 12/04/2012 06:05 PM, Paul van Delst wrote:
> I'm actually on the verge of deciding whether to plunge into all this
> with respect to analysing satellite data simulations. I have to say
> that, given all the Function Graphic MAP related posts over the last
> couple months, GrADS[*] is looking pretty convenient right now.

That's just a personal opinion (I am working in the atmospheric
sciences) but if I had to make a list on how to make a georeferenced
plot look like the way I want it, I would check (in this order):
- cg* routines (if I have an IDL licence)
- NCL http://www.ncl.ucar.edu/
- Python's basemap kit http://matplotlib.org/basemap/users/examples.html

(it's just that GradS plots look a bit ... old ;-))

Fab

Subject: Re: Box Axes with Map Function
Posted by David Fanning on Tue, 04 Dec 2012 18:10:52 GMT
View Forum Message <> Reply to Message

Fab writes:

> That's just a personal opinion (I am working in the atmospheric
> sciences) but if I had to make a list on how to make a georeferenced
> plot look like the way I want it, I would check (in this order):
> - cg* routines (if I have an IDL licence)
> - NCL http://www.ncl.ucar.edu/
> - Python's basemap kit http://matplotlib.org/basemap/users/examples.html
>
> (it's just that GradS plots look a bit ... old ;-))

Oddly, I am trying to learn how to work with Function Graphics map projections because Map_Grid (and, by extension, cgMap_Grid) is driving me CRAZY!!

Map_Grid is designed to work with Map_Set. It is extremely inadequate when you use it with Map_Proj_Init, at least for map projections that don't cover the entire earth.

Since I never use Map_Set, unless under extreme duress, I have had to modify Map_Grid to make it work more adequately with Map_Proj_Init, which I always use to define my map projection space. As long as my map projection does not cover the entire globe, cgMap_Grid works great. As soon as I use it on a global map projection, it screws up. If I fix it to work with global projections, it screws up non-global projections. I'm always poking at the darn thing.

I was hoping that function graphics map commands, because they use Map_Proj_Init (at least I assume they do, since bugs have been fixed in this routine lately), would probably get the grids working correctly. For the most part, this seems to be true.

But, if you total up all the time it has taken me to get a simple map created in what I think of as the "correct" way in function graphics, then I could have probably rewritten Map_Grid 10 times over! And, debugged it, too!

I guess, I might have done this, except that I have a sneaking suspicion that the problem is not in Map_Grid, but in Map_Proj_Init. If you make a map projection in which the longitudes run from -180 to 180 degrees, centered

at 0 degrees longitude, then in projected XY meters, you should have negative values to the left and positive values to the right. Map_Proj_Init will do this correctly, until you get to the right edge of the projected XY space, when the last value will go negative on you to match the value at the left edge.

You could argue that this is the same point, and so the sign of the value doesn't matter. Except that it DOES matter if you are trying to draw a line (think "grid") from one part of the map to the other. That, in essence is the problem I've been trying to solve.

I guess map makers have had this problem for eons, but I still haven't figured out how to solve it correctly in IDL. I'm open to ideas. :-)

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: Box Axes with Map Function
Posted by Matt[2] on Tue, 04 Dec 2012 20:34:00 GMT
View Forum Message <> Reply to Message

Fab <fabien.maussion@gmail.com> writes:

> - Python's basemap kit http://matplotlib.org/basemap/users/examples.html

I can say that we've started using basemap here at NSIDC and have had some luck.

nsidc.org/greenland-today

But we're also still using DG with map_proj*

Cheers,
Matt

--

Matthew Savoie - Senior Software Developer
National Snow and Ice Data Center
(303) 735-0785 http://nsidc.org