Subject: GridData Conundrum
Posted by David Fanning on Sat, 17 Apr 2010 17:19:45 GMT
View Forum Message <> Reply to Message

Folks,

I have long thought that the IDL gridding routine, GridData,
to be one of IDL's most powerful and useful routines. Perhaps
taking its place among the likes of Histogram and Value_Locate.
Well, it *would* be powerful and useful if I could ever
get the damn thing to work. But, alas, I never have been
able to accomplish this simple feat.

I've decided to come clean about my abysmal failure
and ask for your help.

I ran into the perfect test case this week. A simple nearest
neighbor gridding problem that I know how to solve in two
completely independent ways, each producing identical
results. I *know* what I am doing here and I am
supremely confident in the results. "And," I thought,
"it is so simple, I could do this in GridData!"

Not. :-(

I've explained the problem and put some data here on
my web page:

http://www.dfanning.com/code_tips/usegriddata.html

I would be *extremely* grateful to anyone who can take
me by the hand and lead me to the promised land.

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: GridData Conundrum
Posted by David Fanning on Mon, 19 Apr 2010 14:03:05 GMT
View Forum Message <> Reply to Message

Klemen writes:

> Hi David, I have no problems with GRIDDATA; take a look at the code.
> The only problem I had was the triangulate function - you might have
> problems with collinear points on the poles if you don't remove
> them).

Ah, ha! Thanks, Klemen.

This is similar to what I had originally done, but I couldn't overcome that darn collinear points problem. Eventually, I made triangles from the original data XY and that "worked", so I thought that must be right. It never occurred to me that the pole points could be the problem.

Just one correction to your code. You set up a polar stereographic map projection like this:

map=map_proj_init(106, DATUM=8, /GCTP, center_lon=-45., center_lat=90.)

This is incorrect. In the polar stereographic projection (only!!) the CENTER_LAT keyword is actually the TRUE_SCALE_LATITUDE keyword. (This is being fixed, thank goodness, in IDL 8.0, since *everyone* makes this mistake.)

http://www.dfanning.com/map_tips/polarstereo.html

The projection should really be set up like this:

map=map_proj_init(106, DATUM=8, /GCTP, center_lon=-45., center_lat=70.)

I'm really grateful for your help. :-)

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: GridData Conundrum
Posted by Klemen on Mon, 19 Apr 2010 14:44:00 GMT

Hi David,
thank you for TRUE_SCALE_LATITUDE tip, I wasn't aware of it!
Cheers, Klemen

Subject: Re: GridData Conundrum
Posted by David Fanning on Mon, 19 Apr 2010 18:52:19 GMT

Klemen writes:

> Hi David, I have no problems with GRIDDATA; take a look at the code.
> The only problem I had was the triangulate function - you might have
> problems with collinear points on the poles if you don't remove
> them).

I've discovered a couple more interesting facts about this process this morning. It turns out that it is mostly Triangulate that is giving me problems. I've found I do NOT have to exclude any values to produce the proper triangulation, and that the "co-linear" problem occurs on my Windows box, but not my Linux box. On Windows, setting the TOLERANCE keyword to 1 appears to solve the problem.

Also interesting is that there is a small gap (mostly camouflaged in my web article) where the longitude vector wraps around from 257.5 to 0 degrees. This is especially apparent in the filled contour method, and less apparent in the NSIDC regrid method. It appears to disappear completely in the GridData method, perhaps justifying my confidence in its power, if we can just learn to harness it. :-)

I'll update my web page article sometime soon. But managing to do this with GridData opens up a path I have been searching for for at least the last two years!

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 thos speakest truth.")

Subject: Re: GridData Conundrum
Posted by David Fanning on Mon, 19 Apr 2010 20:46:27 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> I've discovered a couple more interesting facts about
> this process this morning.

Just one more surprise to report. If I use the GridData
Natural Neighbor method (instead of the Nearest Neighbor
method), the result is indistiguishable from the Contour
method!

I've updated my web page to report the new results:

http://www.dfanning.com/code_tips/usegriddata.html

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 thos speakest truth.")

Subject: Re: GridData Conundrum
Posted by David Fanning on Wed, 21 Apr 2010 19:23:51 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> Just one more surprise to report. If I use the GridData
> Natural Neighbor method (instead of the Nearest Neighbor
> method), the result is indistiguishable from the Contour
> method!
>
> I've updated my web page to report the new results:
>

> http://www.dfanning.com/code_tips/usegriddata.html

Having GridData working is like living in the land of
milk and honey. But there is one disturbing fact that
lurks like a snake in the garden. I get different
results from GridData depending upon what machine
I run the identical code on. :-(

In particular, the Modifed Shepard's algorithm blows
up (no other way to describe it) on my Windows 64-bit
machine, while it seems to work reasonably well on
my 32-bit LINUX machine.

Can anyone think of any reasonable explanation for this?
I show some results at the end of this article:

   http://www.dfanning.com/code_tips/usegriddata.html

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: GridData Conundrum
Posted by ben.bighair on Thu, 22 Apr 2010 12:35:22 GMT
View Forum Message <> Reply to Message

On Apr 21, 3:23 pm, David Fanning <n...@dfanning.com> wrote:
> David Fanning writes:
>> Just one more surprise to report. If I use the GridData
>> Natural Neighbor method (instead of the Nearest Neighbor
>> method), the result is indistiguishable from the Contour
>> method!
>
>> I've updated my web page to report the new results:
>
>> http://www.dfanning.com/code_tips/usegriddata.html
>
> Having GridData working is like living in the land of
> milk and honey. But there is one disturbing fact that

> lurks like a snake in the garden. I get different
> results from GridData depending upon what machine
> I run the identical code on. :-(
>
> In particular, the Modifed Shepard's algorithm blows
> up (no other way to describe it) on my Windows 64-bit
> machine, while it seems to work reasonably well on
> my 32-bit LINUX machine.
>
> Can anyone think of any reasonable explanation for this?
> I show some results at the end of this article:
>
> http://www.dfanning.com/code_tips/usegriddata.html
>

Hi David,

It doesn't look to me like you are running the coordinates through GRID_INPUT before passing them to GRIDDATA. In the past I found that (generally) removed all subsequent potholes from the GRIDDATA process.

Cheers,
Ben

Subject: Re: GridData Conundrum
Posted by David Fanning on Thu, 22 Apr 2010 13:07:35 GMT
View Forum Message <> Reply to Message

ben.bighair writes:

> It doesn't look to me like you are running the coordinates through
> GRID_INPUT before passing them to GRIDDATA. In the past I found that
> (generally) removed all subsequent potholes from the GRIDDATA process.

Well, I tried this:

```
Grid_Input, x, y, air, x1, y1, air1
Triangulate, x1, y1, triangles1, TOLERANCE=1.0
griddedData = GRIDDATA(x1, y1, air1, $
    /SHEPARDS, DELTA=[25000.,25000.], TRIANGLES=triangles1, $
    DIMENSION=[304,448], START=[-3850000., -5350000.])
```

Now the Shepard's plot "blows up", but in a more orderly fashion, with straight lines. :-)

Still works on my LINUX box, though, which I find most disconcerting.

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.")
