
Subject: GRIDDATA woes

Posted by [ben.bighair](#) on Mon, 03 Mar 2008 02:57:27 GMT

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

Hi All,

I have been having a problem similar to this one... <http://tinyurl.com/2spe3v>

The solution to the problem in the above posting was to use GRID_INPUT to filter and reorder the data *before* calling QHULL and GRIDDATA.

That doesn't seem to be the case this time as I faithfully perform these steps. However, the error message indicates that it is something similar is going on.

The big picture is that I have an irregular grid (actually it is regular in longitude but irregular in latitude) that I want to interpolate onto a regular grid. I have assembled a mockup of the situation in this procedure...

http://www.tidewater.net/~pemaquid/counterclockwise_fail.pro

The error message when the above is run is ...

```
SeaDAS> z=counterclockwise_fail()
% GRIDDATA: Triangle 5 not in counterclockwise order.
% GRIDDATA: Triangle 6 not in counterclockwise order.
% GRIDDATA: Triangle 7 not in counterclockwise order.
% GRIDDATA: Triangle 17 not in counterclockwise order.
% GRIDDATA: Triangle 30 not in counterclockwise order.
% GRIDDATA: Triangle 31 not in counterclockwise order.
% GRIDDATA: Triangle 34 not in counterclockwise order.
% GRIDDATA: Triangle 40 not in counterclockwise order.
% GRIDDATA: Triangle 42 not in counterclockwise order.
% GRIDDATA: Triangle 49 not in counterclockwise order.
```

I have tried changing the values in the code to double. That results in a similar set of errors but for a different set of triangles.

```
% GRIDDATA: Triangle 4 not in counterclockwise order.
% GRIDDATA: Triangle 5 not in counterclockwise order.
% GRIDDATA: Triangle 6 not in counterclockwise order.
% GRIDDATA: Triangle 16 not in counterclockwise order.
% GRIDDATA: Triangle 33 not in counterclockwise order.
% GRIDDATA: Triangle 35 not in counterclockwise order.
% GRIDDATA: Triangle 36 not in counterclockwise order.
% GRIDDATA: Triangle 39 not in counterclockwise order.
% GRIDDATA: Triangle 42 not in counterclockwise order.
% GRIDDATA: Triangle 45 not in counterclockwise order.
```

Bah!

I have seen a number of messages on the newsgroup about interpolation from an irregular grid to a regular one. None appear to address the issues around gridding on a sphere. I don't think I can use anything as simple as INTERPOLATE since the input array is sampled at irregular intervals.

So how is this kind of interpolation supposed to be done?

Thanks!

Ben

** Structure !VERSION, 8 tags, length=76, data length=76:

```
ARCH      STRING  'ppc'
OS         STRING  'darwin'
OS_FAMILY  STRING  'unix'
OS_NAME    STRING  'Mac OS X'
RELEASE    STRING  '6.3'
BUILD_DATE STRING  'Mar 23 2006'
MEMORY_BITS INT     32
FILE_OFFSET_BITS
            INT     64
```

Subject: Re: GRIDDATA woes

Posted by [David Fanning](#) on Tue, 04 Mar 2008 23:58:42 GMT

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

Bill Gallery writes:

- > On the other hand, I don't understand the implications of the /sphere
- > option: without it, you appear to be interpolating on a flat surface.
- > What happens near the poles? What about crossing the meridian from 359
- > deg to 0 deg? I never investigated these question. Perhaps you can
- > elucidate them in your white paper.

Yes, exactly my plan, but at the moment it looks like I will have to re-write GRIDDATA from scratch in order to understand it. :-(

I always get an ominous feeling when I've asked the same question three or four times and get nothing but silence back. An obvious topic for a Ph.D. thesis, is my first thought. In this case, I was hoping the author might be listening in and could tell us why he added the darn SPHERE keyword in the first place. Surely he had something other than confounding users in mind. :-)

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 woes

Posted by [ben.bighair](#) on Wed, 05 Mar 2008 03:14:26 GMT

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

On Mar 4, 6:58 pm, David Fanning <n...@dfanning.com> wrote:

> Bill Gallery writes:

>> On the other hand, I don't understand the implications of the /sphere

>> option: without it, you appear to be interpolating on a flat surface.

>> What happens near the poles? What about crossing the meridian from 359

>> deg to 0 deg? I never investigated these question. Perhaps you can

>> elucidate them in your white paper.

>

> Yes, exactly my plan, but at the moment it looks like I will have to

> re-write GRIDDATA from scratch in order to understand it. :-(

>

> I always get an ominous feeling when I've asked the same

> question three or four times and get nothing but silence

> back. An obvious topic for a Ph.D. thesis, is my first

> thought. In this case, I was hoping the author might be

> listening in and could tell us why he added the darn

> SPHERE keyword in the first place. Surely he had something

> other than confounding users in mind. :-)

>

Hi,

I updated the example code using Bill's suggestions and a small twist...

http://www.tidewater.net/~pemaquid/ccw_fail.pro

This test function now allows you to specify the triangulation method ("qhull" or "triangulate") and the /SPHERE keyword. Here's the results using the different keyword combinations...

SeaDAS> z = ccw_fail(olon = olon, olat = olat, trimethod = "qhull",
sphere = 0) ;works

```
SeaDAS> z = ccw_fail(olon = olon, olat = olat, trimethod = "qhull",
sphere = 1) ;doesn't work
% GRIDDATA: Triangle 5 not in counterclockwise order.
% GRIDDATA: Triangle 6 not in counterclockwise order.
```

```
.
.
.
```

```
SeaDAS> z = ccw_fail(olon = olon, olat = olat, trimethod =
"triangulate", sphere = 0) ;works
```

```
SeaDAS> z = ccw_fail(olon = olon, olat = olat, trimethod =
"triangulate", sphere = 1) ;works
```

QHULL and TRIANGULATE use different algorithms, so I suppose it is reasonable that they produce different outputs. So, I compared the results of each and sure enough they are different...

```
QHULL, lon, lat, qtr, /DELAUNAY
TRIANGULATE, lon, lat, ttr, /DEGREES
help, qtr, ttr
QTR      LONG    = Array[3, 744557] ;QHULL's triangulation
TTR      LONG    = Array[3, 742122] ;TRIANGULATES
triangulation
```

So, I am not sure what to think. The locations are specified to be on a grid it is unlikely that the issue is duplicates points (within some tolerance). For grins I called QHULL with the SPHERE keyword assigned...

```
QHULL, lon, lat, sqtr, /DELAUNAY, SPHERE = s
help, sqtr
SQTR      LONG    = Array[3, 744578]
```

Hahaha! Time to call it quits for the day!

Cheers,
Ben

Subject: Re: GRIDDATA woes
Posted by [David Fanning](#) on Wed, 05 Mar 2008 05:04:32 GMT
[View Forum Message](#) <> [Reply to Message](#)

Kenneth P. Bowman writes:

> Sad to say, I refer to the book more often than I would like to,

> but at least I have written a lot of things down where I can find
> them. :-)

The *exact* purpose of my web page, although I try to
let on to people that I do it for them. :-)

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 woes
Posted by [David Fanning](#) on Wed, 05 Mar 2008 05:10:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

ben.bighair writes:

> For grins I called QHULL with the SPHERE keyword
> assigned...
>
> QHULL, lon, lat, sqtr, /DELAUNAY, SPHERE = s
> help, sqtr
> SQTR LONG = Array[3, 744578]
>
> Hahaha! Time to call it quits for the day!

I just got back from a David Quammen lecture
(Song of the Dodo is one of my all-time favorite
science books, so I had him sign my copy), and I am
too tired to think about this. But I have the
feeling I am going to be awake all night long. :-(

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 woes
Posted by [jameskuyper](#) on Wed, 05 Mar 2008 11:38:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

Kenneth P. Bowman wrote:

...

- > Remember, interpolation is an approximation. You make assumptions about the
- > behavior of a function between known (tabulated) points. Bilinear
- > interpolation is the crudest possible interpolation scheme.

That depends upon whether you're willing to call nearest neighbor an interpolation method. I think that it is, in the same sense that a point can be considered a degenerate case of a line segment, with a length of 0.

Subject: Re: GRIDDATA woes
Posted by [Kenneth P. Bowman](#) on Wed, 05 Mar 2008 13:46:41 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <CCvzj.10558\$1_.7720@trnddc02>,
James Kuyper <jameskuyper@verizon.net> wrote:

- > Kenneth P. Bowman wrote:
- > ...
- >> Remember, interpolation is an approximation. You make assumptions about the
- >> behavior of a function between known (tabulated) points. Bilinear
- >> interpolation is the crudest possible interpolation scheme.
- >
- > That depends upon whether you're willing to call nearest neighbor an
- > interpolation method. I think that it is, in the same sense that a point
- > can be considered a degenerate case of a line segment, with a length of 0.

I knew someone would point that out. ;-)

OK then, second-crudest approximation.

Ken

Subject: Re: GRIDDATA woes
Posted by [David Fanning](#) on Wed, 05 Mar 2008 14:57:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

Kenneth P. Bowman writes:

- > VALUE_LOCATE finds the index of the point less than or equal to the search
- > value. You are trying to interpolate exactly to the last point. This
- > code correctly computes the index of that point to be 7, but there

```

> is no point 8 to use for the interpolation. This can be solved
> like this
>
> IDL> lat = [-87.5, 50, 25, 0, 30, 45, 64, 87.5]
> IDL> y = Scale_Vector(findgen(7), -87.5, 87.499)  <-----
> IDL> j = Value_Locate(lat, y)
> IDL> yj = j + (y - lat[j])/(lat[j+1] - lat[j])
> IDL> PRINT, yj
>    0.00000    0.212120    0.424240    0.636360    3.97220
>    5.70171    6.99996
>
> Unfortunately, INTERPOLATE does not extrapolate when you are outside
> the domain of the function.

```

I'm going to agree with Ben, and keep VALUE_LOCATE out of it. But, I've also used Paolo's suggestion. Here is code that I think works well for me, and allows me to create any sized grid I want.

In the code, lat is a 48-element vector that is irregularly spaced, lon is a 96-element vector, that is regularly spaced, except for the two end members, and sit is the 2D array I wish to resample. In this code, I am trying to resample to a 360x180 array, to make it consistent with other arrays I have available to me.

```

nx = 360
ny = 180
slon = Scale_Vector(Findgen(nx), 0.5, 359.5)
slat = Scale_Vector(Findgen(ny), Min(lat), Max(lat))
x = Interpol(Findgen(N_Elements(lon)), lon, slon)
y = Interpol(Findgen(N_Elements(lat)), lat, slat)
xx = Rebin(x, nx, ny, /SAMPLE)
yy = Rebin(Reform(y, 1, ny), nx, ny)

```

```

resampledArray = Interpolate(sit, xx, yy)

```

Even in this "worst case" scenario, I seem to get reasonably good results. I *really* like that INTERPOL way of getting fractional indices!

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.

Subject: Re: GRIDDATA woes
Posted by [Kenneth Bowman](#) on Wed, 05 Mar 2008 16:05:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <MPG.223862e4a9bdefba98a2b0@news.frii.com>,
David Fanning <news@dfanning.com> wrote:

```
> In the code, lat is a 48-element vector that is irregularly
> spaced, lon is a 96-element vector, that is regularly
> spaced, except for the two end members, and sit is the
> 2D array I wish to resample. In this code, I am trying to
> resample to a 360x180 array, to make it consistent with other
> arrays I have available to me.
>
> nx = 360
> ny = 180
> slon = Scale_Vector(Findgen(nx), 0.5, 359.5)
> slat = Scale_Vector(Findgen(ny), Min(lat), Max(lat))
> x = Interpol(Findgen(N_Elements(lon)), lon, slon)
> y = Interpol(Findgen(N_Elements(lat)), lat, slat)
> xx = Rebin(x, nx, ny, /SAMPLE)
> yy = Rebin(Reform(y, 1, ny), nx, ny)
```

Usually these grids are (360 x 181), with longitudes from [0, 359] and latitudes from [-90, 90], but you may have specific reasons for wanting your particular set of lats and lons.

Ken
