Subject: How to grid pixel level data where latitude and longitude are 2D arrays
Posted by masterjedirobyn on Wed, 19 Jun 2013 19:31:06 GMT
View Forum Message <> Reply to Message

Hi,

I have been having a problem gridding a very large dataset which contains pixel level data into a gridded average. My data looks like this:

Lat and Lon are float [409,13248], the variables I wish to interpolate are also [409,13248]. I wish to grid these into arrays of [360,180] (1 degree spacing). Lon and Lat are irregular.

I have tried several methods of doing this. First, I looked at http://www.idlcoyote.com/code_tips/griddata.html and followed the process there, using qhull. Then, using the griddata command gave the following error: GRIDDATA: Value of Triangle index is out of allowed range. I then used triangulate instead of qhull and did not run into an error, but the result I got does not seem to be correct. Even if it was correct, the amount of time this calculation takes is huge; it runs overnight, and that's only on one file. I have many.

Does anyone know a faster, more memory efficient way of gridding data when your latitude and longitude are irregular and in 2D? This calculation typically freezes my machine with IDL using over 100% of the CPU. Is it possible that there could be some trick using Value_Locate? (Although from what I read, value_locate only works when your lon/lats are monotonically increasing/decreasing)

Thanks

Subject: Re: How to grid pixel level data where latitude and longitude are 2D arrays
Posted by David Fanning on Wed, 19 Jun 2013 20:30:56 GMT
View Forum Message <> Reply to Message

masterjedirobyn@gmail.com writes:

> I have been having a problem gridding a very large dataset which contains pixel level data into a gridded average. My data looks like this:
>
> Lat and Lon are float [409,13248], the variables I wish to interpolate are also [409,13248]. I wish to grid these into arrays of [360,180] (1 degree spacing). Lon and Lat are irregular.
>
> I have tried several methods of doing this. First, I looked at http://www.idlcoyote.com/code_tips/griddata.html and followed the process there, using qhull. Then, using the griddata command gave the following error: GRIDDATA: Value of Triangle index is out of allowed range. I then used triangulate instead of qhull and did not run into an error, but the result I got does not seem to be correct. Even if it was correct, the amount of time this calculation takes is huge; it
runs overnight, and that's only on one file. I have many.
>

> Does anyone know a faster, more memory efficient way of gridding data when your latitude and longitude are irregular and in 2D? This calculation typically freezes my machine with IDL using over 100% of the CPU. Is it possible that there could be some trick using Value_Locate? (Although from what I read, value_locate only works when your lon/lats are monotonically increasing/decreasing)

Well, I have never known GridData NOT to take a lot of time! Although all night does seem a bit excessive. But, I have managed to get it to work, on occasion. Switching from Triangulate to QHull or visa versa rings some bells. Have you tried running Grid_Input on your data first? Have you used the Tolerance keyword to Triangulate? Have you read this article:

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

That is an awful lot of data points. I can see why there is some thrashing going on. Can you get this to work if you take some reasonably small number of random points from your data and worked with those? Maybe you are so oversampled, it won't make any difference. :-)

Cheers,

David

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

Subject: Re: How to grid pixel level data where latitude and longitude are 2D arrays
Posted by David Fanning on Wed, 19 Jun 2013 20:41:29 GMT

View Forum Message <> Reply to Message

David Fanning writes:

> That is an awful lot of data points. I can see why there is some
> thrashing going on. Can you get this to work if you take some reasonably
> small number of random points from your data and worked with those?
> Maybe you are so oversampled, it won't make any difference. :-)

What if you used HIST_ND to bin up your lat/lon arrays, then looped through each bin and used the reverse indices vector to select the data values you want to use in each bin. Take the median value of the data values as the value for the bin. That would take seconds, rather than days.

Cheers,

David

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

Subject: Re: How to grid pixel level data where latitude and longitude are 2D arrays
Posted by masterjedirobyn on Wed, 19 Jun 2013 20:58:57 GMT

View Forum Message <> Reply to Message

On Wednesday, June 19, 2013 4:41:29 PM UTC-4, David Fanning wrote:
> David Fanning writes:
>
>
>
>> That is an awful lot of data points. I can see why there is some
>
>> thrashing going on. Can you get this to work if you take some reasonably
>
>> small number of random points from your data and worked with those?
>
>> Maybe you are so oversampled, it won't make any difference. :-)
>
>
>
> What if you used HIST_ND to bin up your lat/lon arrays, then looped
>
> through each bin and used the reverse indices vector to select the data
>
> values you want to use in each bin. Take the median value of the data
>
> values as the value for the bin. That would take seconds, rather than
>
> days.
>
>
>
> Cheers,
>
>
>
> David
>

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

I am not familiar with the HIST_ND routine, but I have used hist_2d to make frequency density plots before. I'm having trouble wrapping my head around what the call to hist_nd would be. The syntax is

hist=HIST_ND(V,[BINSIZE,MIN=,MAX=,NBINS=,REVERSE_INDICES=])

and I have lat[409,13248],lon[409,13248],var[409,13248]. Would I call something like this:

hist_lat=hist_nd(lat,binsize=1,min=-90,max=90,reverse_indice s=ri_lat)
and
hist_lon=hist_nd(lon,binsize=1,min=-180,max=180,reverse_indi ces=ri_lon)

and then I would loop through -90 to 90 for lat and select the median from the bin (and -180 to 180 for lon), which would leave me with 1D lat and lon arrays? I could then use these arrays with an interpolate command, thus avoiding griddata altogether? I apologize if I'm completely wrong in how I understand this.

As per your previous reply, I am currently running grid_input on my data, but it's been running for several hours. I have never tried using a tolerance keyword in triangulate, but I may try after grid_input finishes running. After reading the article above on usegriddata.html, is the key part of that article the use of the map_proj routines? (i.e., using map_proj_init and map_proj_forward on the lats and lons before passing them to triangulate?)

Thank you so much for your replies.

Subject: Re: How to grid pixel level data where latitude and longitude are 2D arrays
Posted by David Fanning on Wed, 19 Jun 2013 21:17:50 GMT
View Forum Message <> Reply to Message

masterjedirobyn@gmail.com writes:

> I am not familiar with the HIST_ND routine, but I have used hist_2d to
make frequency density plots before. I'm having trouble wrapping my head
around what the call to hist_nd would be. The syntax is
>

> hist=HIST_ND(V,[BINSIZE,MIN=,MAX=,NBINS=,REVERSE_INDICES=])
>
> and I have lat[409,13248],lon[409,13248],var[409,13248]. Would I call something like this:
>
> hist_lat=hist_nd(lat,binsize=1,min=-90,max=90,reverse_indice s=ri_lat)
> and
> hist_lon=hist_nd(lon,binsize=1,min=-180,max=180,reverse_indi ces=ri_lon)
>
> and then I would loop through -90 to 90 for lat and select the median from the bin (and -180 to 180 for lon), which would leave me with 1D lat and lon arrays? I could then use these arrays with an interpolate command, thus avoiding griddata altogether? I apologize if I'm completely wrong in how I understand this.

Well, Hist_ND is what Hist_2D was aspiring to be. :-)

Hist_ND is JD Smith's routine (and so, written extremely well). If you
can't find a copy on his web page, you can find a copy (probably older)
in the Public folder of the Coyote Library. The real reason to use it
here is that it returns the reverse indices for you. Hist_2D doesn't do
that.

http://www.idlcoyote.com/programs/public/hist_nd.pro

I would bin your lat and lon arrays (at the same time!) using Hist_ND.
Then, I would loop through each bin (360*180 of them), using the indices
for that bin to select the data values you want to use in calculating
the single data value for that bin. I suppose you can do this part in
various ways, but I would start by just getting the median value, I
think.

> After reading the article above on usegriddata.html, is the key part
> of that article the use of the map_proj routines? (i.e., using
> map_proj_init and map_proj_forward on the lats and lons before
> passing them to triangulate?)

I don't know if that is the "key part", but I can't get my head around
anything but rectangular grids, especially when it comes to map
projections, so I do EVERYTHING in XY space, not lat/lon space. At least
then I can explain what I am doing to someone. And, yes, it makes it
easier to form triangles when the points are not all bunched up in the
same location. :-)

Cheers,

David

--

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

Subject: Re: How to grid pixel level data where latitude and longitude are 2D arrays
Posted by dplatten on Thu, 20 Jun 2013 11:00:40 GMT

View Forum Message <> Reply to Message

Hi David,

I've just read your reply about using HIST_ND instead of GRIDDATA. As a result I've rewritten a bit of GRIDDATA code that I've been using so that it now uses HIST_ND and the reverse indices. GRIDDATA used to take at least ten minutes to process my data files, whereas the new HIST_ND version takes about a second. Many thanks for your post!

David

On Wednesday, June 19, 2013 10:17:50 PM UTC+1, David Fanning wrote:
>
>
> Hist_ND is JD Smith's routine (and so, written extremely well). If you
>
> can't find a copy on his web page, you can find a copy (probably older)
>
> in the Public folder of the Coyote Library. The real reason to use it
>
> here is that it returns the reverse indices for you. Hist_2D doesn't do
>
> that.
>
>
>
>   http://www.idlcoyote.com/programs/public/hist_nd.pro
>
>
>
> I would bin your lat and lon arrays (at the same time!) using Hist_ND.
>
> Then, I would loop through each bin (360*180 of them), using the indices
>
> for that bin to select the data values you want to use in calculating
>
> the single data value for that bin. I suppose you can do this part in
>
> various ways, but I would start by just getting the median value, I

>
> think.
>
>
>
> I don't know if that is the "key part", but I can't get my head around
>
> anything but rectangular grids, especially when it comes to map
>
> projections, so I do EVERYTHING in XY space, not lat/lon space. At least
>
> then I can explain what I am doing to someone. And, yes, it makes it
>
> easier to form triangles when the points are not all bunched up in the
>
> same location. :-)
>
>
>
> Cheers,
>
>
>
> David
>
>
>
> --
>
> David Fanning, Ph.D.
>
> Fanning Software Consulting, Inc.
>
> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
>
> Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: How to grid pixel level data where latitude and longitude are 2D arrays
Posted by David Fanning on Thu, 20 Jun 2013 12:35:33 GMT

View Forum Message <> Reply to Message

David Platten writes:

> I've just read your reply about using HIST_ND instead of GRIDDATA. As a result I've rewritten a bit of GRIDDATA code that I've been using so that it now uses HIST_ND and the reverse indices. GRIDDATA used to take at least ten minutes to process my data files, whereas the new HIST_ND version takes about a second. Many thanks for your post!

Really!? OK, I'm done with the one good idea I try to come up with every week. Guess I'll go fishing. ;-)

Cheers,

David

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