
Subject: Re: spatial interpolation

Posted by [Mark Hadfield](#) on Sat, 27 Sep 2003 23:13:22 GMT

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

Isa Usman wrote:

> Hello All,

>

> I have a program which interpolates an irregularly gridded set of data
> points onto another irregular grid. I have tried as much as possible to make
> the calculations as fast as possible (using the dreaded reverse indices in
> Histogram) but i am at my wits end. It currently takes about two days to go
> through the whole data. Anybody got any suggestions on speed-up
> improvements? The code is shown below.

What do you mean by "irregularly gridded"? (Sorry, but I can't determine this from your code.) Are your data points randomly scattered about, or are they on some sort of deformed, stretched, or rotated Cartesian grid? Or something else?

If you do have two grids (taking the word to mean a set of nodes with some sort of geometric structure) then the key part of your regridding is to determine where the nodes of the first grid are relative to the nodes of the second. I have some routines to do this for 2D curvilinear grids, one using triangular linear interpolation and the other using Powell minimisation. I can explain further or send you the code, but first I need to know more about what you are trying to do.

--

Mark Hadfield "Ka puwaha te tai nei, Hoesa tatou"

m.hadfield@niwa.co.nz

National Institute for Water and Atmospheric Research (NIWA)

Subject: Re: spatial interpolation

Posted by [Isa Usman](#) on Mon, 29 Sep 2003 11:57:49 GMT

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

"Mark Hadfield" <m.hadfield@niwa.co.nz> wrote in message
news:bl55ik\$iq9\$1@newsreader.mailgate.org...

> Isa Usman wrote:

>> Hello All,

>>

>> I have a program which interpolates an irregularly gridded set of data
>> points onto another irregular grid. I have tried as much as possible to make
>> the calculations as fast as possible (using the dreaded reverse indices
in

>> Histogram) but i am at my wits end. It currently takes about two days to go
>> through the whole data. Anybody got any suggestions on speed-up
>> improvements? The code is shown below.
>
> What do you mean by "irregularly gridded"? (Sorry, but I can't determine
> this from your code.) Are your data points randomly scattered about, or
> are they on some sort of deformed, stretched, or rotated Cartesian grid?
> Or something else?
>
> If you do have two grids (taking the word to mean a set of nodes with
> some sort of geometric structure) then the key part of your regridding
> is to determine where the nodes of the first grid are relative to the
> nodes of the second. I have some routines to do this for 2D curvilinear
> grids, one using triangular linear interpolation and the other using
> Powell minimisation. I can explain further or send you the code, but
> first I need to know more about what you are trying to do.

Sorry, I should have really said that the data I am interpolating from (radar data) is on a polar grid. But because the data does not have a central node due to only data within a radial distance of 20km and 40km being available, I termed it as irregularly gridded. The data spans out in ~0.25 degree increments up to an angle of 50 degrees. The points I am interpolating to have a central node situated on the centre of the plane defined by the radar data points.

What I did in the program was to histogram the original points and the points that I wanted to interpolate to over a certain rectangular area. Essentially this constructed a mesh grid over the points and then it would loop over each grid to do the interpolation. I did this so that i wouldn't need to loop over every point to interpolate. To make sure there weren't any "edge effects" in the interpolation, either 8 or 24 grids surrounding the main grid were joined together before interpolating using MIN_CURVE_SURF.

I hope this helps...

Isa

Subject: Re: spatial interpolation

Posted by [Mark Hadfield](#) on Tue, 30 Sep 2003 00:40:49 GMT

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

Isa Usman wrote:

> Sorry, I should have really said that the data I am interpolating from
> (radar data) is on a polar grid. But because the data does not have a
> central node due to only data within a radial distance of 20km and 40km
> being available, I termed it as irregularly gridded. The data spans out in

- > ~0.25 degree increments up to an angle of 50 degrees. The points I am
- > interpolating to have a central node situated on the centre of the plane
- > defined by the radar data points.
- >
- > What I did in the program was to histogram the original points and the
- > points that I wanted to interpolate to over a certain rectangular area.
- > Essentially this constructed a mesh grid over the points and then it would
- > loop over each grid to do the interpolation. I did this so that i wouldn't
- > need to loop over every point to interpolate. To make sure there weren't any
- > "edge effects" in the interpolation, either 8 or 24 grids surrounding the
- > main grid were joined together before interpolating using MIN_CURVE_SURF.

Hmm, I think I understand that. But I'm afraid understanding how other people do things is **really** not my strong point, so I'll just set down some thoughts and principles that might be helpful.

- The best IDL procedure for interpolating from irregularly gridded data is GRIDDATA, introduced in 5.5. Notwithstanding what it says in the first paragraph of the documentation, it can interpolate to any output grid, irregular or otherwise.
- When you have good data coverage and you want to interpolate within the area bounded by the data points, the most robust method, if not the most accurate, is probably linear interpolation.
- Linear interpolation with GRIDDATA requires TRIANGLES data to be supplied (as do several other methods). The TRIANGLES data defines how to connect the points before interpolating. GRIDDATA then determines which triangle each output point is contained in and interpolates linearly to it from the three vertices.
- For scattered input data you can generate a triangulation with the TRIANGULATE routine. This can be quite time consuming. If your input data are on a regular grid then you can construct a triangulation to suit the grid. For example, I have a routine that constructs a triangulation for a rectangular mesh.
- GRIDDATA can also carry out minimum curvature surface interpolation, as does the older MIN_CURVE_SURF fuction. This method is very expensive on large data sets: execution time increases with the cube of the number of input data points. You can avoid this expense by subdividing your domain (as you seem to be doing), but GRIDDATA also allows you to cut execution time drastically by specifying the MIN_POINTS keyword.
- In my line of business (ocean hydrodynamic modelling) I interpolate a lot between rectilinear and curvilinear grids (the latter being a mesh of rectangular cells like the former, but rotated or distorted). I like to do this as a two-stage process: 1) locate each output data point in

the "index space" of the input grid; 2) do the interpolation with INTERPOLATE. There are a couple of reasons for doing it this way: 1) it is cheaper than using GRIDDATA if I want to carry out several interpolations on the same pair of grids; 2) missing data can be handled better. My Motley library at <http://www.dfanning.com/hadfield/README.html> has examples of this approach.

Good luck!

--

Mark Hadfield "Ka puwaha te tai nei, Hoesa tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)

Subject: spatial interpolation
Posted by [hjalti](#) on Wed, 01 Oct 2003 20:45:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello

Has anyone interpolated values from one set of scattered points (say the terrain height of digitized contour lines from a map) to another set of scattered points (say those points that are to be used in a finite element model for groundwater flow). Note that this does not involve any type of grids, regular nor irregular.
What would be the most straightforward way to do this?

Regards, Hjalti
