Subject: Re: GRIDDATA woes
Posted by Kenneth P. Bowman on Mon, 03 Mar 2008 14:24:14 GMT

View Forum Message <> Reply to Message

In article
<57808cc6-8454-45f1-a104-50e465ef294c@v3g2000hsc.googlegroups.com>,
"ben.bighair" <ben.bighair@gmail.com> wrote:

> 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?

If your grid is rectangular and separable (in the sense that all the longitudes in each "column" of data are the same and all of the latitudes in each "row" of data are same), even if the coordinates are not regularly spaced, then it is actually quite easy to interpolate to any set of points (regular or irregular) using INTERPOLATE. This should be much faster than triangulating.

This problem looks just like the one David Fanning was working on recently, and here is an outline of the solution

> Assuming that your data is 2-D (x = longitude and y = latitude), create
> the grids that you want to interpolate to
> nx = 360
> ny = 181

> Compute the "interpolation coordinates" from the original grid

> yj = j + (y - y_original[j])/(y_original[j+1] - y_original[j])
> Since the input and output grids are the same in the x-direction, you

> xx = REBIN(x, nx, ny, /SAMPLE)
> yy = REBIN(REFORM(yi, 1, ny), nx, ny, /SAMPLE)
> Then interpolate
> new = INTERPOLATE(original, xx, yy)

By happy chance, the interpolation chapter from my book is the sample that is posted online here

http://idl.tamu.edu/Book.html

Ken Bowman