Subject: Re: weird behavior of Triangulate
Posted by Yngvar Larsen on Fri, 14 Sep 2012 07:07:06 GMT
View Forum Message <> Reply to Message

On Wednesday, 12 September 2012 16:42:17 UTC+2, David Fanning wrote:
> Yngvar Larsen writes:
>
>
>
>> ;; xcoord_in/ycoord_in are the coordinates of the grid points of your input grid.
>> ;; dx_in/dy_in is the grid resolution in x/y direction.
>> xind = (utm33[0,*] - xcoord_in[0,0])/dx_in
>> yind = (utm33[1,*] - ycoord_in[0,0])/dy_in

[Edit: utm34 -> utm33 in the previous line.]

>> ;; or
>> ;; yind = (ycoord_in[0,0] - coord[1,*])/dy_in
>> ;; if first line of the input data array is "upper row"/"northernmost row" like in your example
>
> My biggest problem is figuring out how to get index
> arrays. I seem to have a mental block against figuring
> it out. As I pondered the problem yesterday, I discovered
> that I could use Scale_Vector to create the index arrays.
> Since I *do* understand Scale_Vector, this has helped
> tremendously.

Well it *is* a "scale_vector" operation, so applied correctly, I'm sure SCALE_VECTOR might ease the cognitive load.

> I still get confused about the index values for latitudes. Do they have to get reversed or
> not!? Maybe not, if I already reversed the data... etc.

That's what my comment in the code snippet above is about. ENVI and many other GIS-like software packages and data formats like to have [0,0] in the upper left corner, while IDL doesn't. ENVI also starts counting from [1,1] for some very strange reason. I really don't want to see the ENVI source code! Imagine implementing indexing from [1,1] in a language that doesn't...

Personally, I always reverse the data when I read from disk, such that [0,0] is in the lower left corner the IDL way. I can then treat x and y coordinates the same way.

--
Yngvar