

---

Subject: Re: TRIANGULATE/TRIGRID problem in IDL 5.3 (SGI)

Posted by [Liam E. Gumley](#) on Fri, 20 Apr 2001 19:46:47 GMT

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

---

Ben Tupper wrote:

> I have bumped into this a number of times. You probably already know this, but  
> IDL's TRIGRID routine interpolates out to the limits of the 'outer-hull'  
> defined by the Delaunay triangulation. The curviness of the data swath  
> (almost banana-shaped) has introduced a subtle concavity. There is a triangle  
> defined to connect the ends of the banana shape. The table shown in the online  
> help of TRIGRID indicates that data can be extrapolated beyond the triangles  
> (which I interpret to mean that interpolation always occurs within the  
> triangles.) The greatest difference between the data locations and the  
> boundary occurs in the middle of the banana. Try changing the TRIANGULATE  
> statement to the following to retrieve the boundary points.

>  
> triangulate, lon[loc], lat[loc], tri, bounds

>  
> Then after you display the image, overplot the boundary points:

>  
> plots, lon[bounds], lat[bounds], psym = -1, color = !P.color

>  
> Note that many boundary points lie along the top of the swath, but few along  
> the bottom (within the swath concavity.)

>  
> Generally, I have post-masked the data just as Craig suggests here. I usually  
> have only a couple hundred (at most) columns and rows and the concavities are  
> not so subtle as yours - so I manually mask out the nonsense data. That  
> doesn't seem practical for your situation. Your data comes in sets of 10  
> (should it be decades or decaduplets?); is each set a scan line? If so,  
> perhaps you could assemble the extremes from each scan line to use as the  
> masking boundary.

>  
> I tried the following and it seems to yield the appropriate boundary points to  
> use for masking.

>  
> lon\_b = lon[\*, [0,9]]  
> lat\_b = lat[\*, [0,9]]  
> ;plot each end of extremes separately  
> plots, lon\_b[\*,0], lat\_b[\*,0], psym = -3, color = 255  
> plots, lon\_b[\*,1], lat\_b[\*,1], psym = -3, color = 200

>  
> Now all you have to do is convert those longitudes and latitudes to image  
> coordinates and feed them to the POLYFILLV function to get the image indices  
> to preserve, the rest gets masked.

Ben,

I already tried the masking strategy you suggested yesterday, and the results weren't satisfactory.

However your suggestion for plotting the boundary nodes got me thinking. The online documentation for TRIGRID shows how you can plot each triangle in the connectivity list, e.g.

```
window, /free, xsize=1100, ysize=850
device, decomposed=0
restore, 'latlon.xdr'
map_set, 30, -112.5, scale=0.2e6
triangulate, lon, lat, tri, b
for i= 0L, n_elements(tri) / 3L - 1L do begin $
t = [tri[*], i], tri[0, i]] & plots, lon[t], lat[t] & endfor
plots, lon, lat, psym=2
```

This shows that many triangles are created along the bottom edge of the swath where I don't want triangles! However it also demonstrates that the triangles are created in a very consistent manner for the swath itself. The swath dimensions of 1354 x 10 are never going to change (yes, it is MODIS data), and I don't believe the connectivity list will ever change. So I can probably compute a triangle list manually just once (without using TRIANGULATE), and then use it for all cases in conjunction with TRIGRID. I'll send an update when I've implemented and tested this idea.

Cheers,  
Liam.

PS Thanks to the other posters for the encouragement and suggestions.

---