Subject: Re: spherical gridding
Posted by Ben Tupper on Sun, 17 Sep 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Hello,

The XGRID and YGRID keywords to TRIGRID will return vectors of the grid
row and column locations. That is, for example, each element of XGRID
will identify the coordinates of each column of the output grid (in your
case degrees longitude).   You might consider using IDLgrSurface which
is set up to handle Z (2d array), and X and Y (vector coordinates for
grid elements).

Ben

Sylvain Carette wrote:

> Not sure if my understanding of spherical gridding correspond to the
> actual implementation in TRIANGULATE and TRIGRID function and proc.
> Running the sample code below as surface give a sine shaped surface
> while I was expecting a sphere.
> I figure that perhaps SURFACE only display "flat" surface and what I
> see is a unwarped sphere (?).
> I though it would be better to feed the trigrid output to a polygon as
> in IDLgrPolygon but now this one want xyz value while as far I can see
> trigrid result is 2 dimentional array...
> What do I miss?
>
> My intend here is to project elevations grids onto wgs84 geoid but
> projecting onto a perfect sphere will be just fine to begin with. If
> I'm unclear, I need the final dem be patch of a sphere so that when
> putting all of them together they form a globe and the cartesian space
> holding it have its origin at the center of the globe.
>
> I've done once an autolisp function in Autocad doing this but the dem
> have to be converted first to dxf in no more than 300x300 chunk and it
> do not handle the case for dems that are not in long/lat proj. Very,
> very big and slow...
>
> Here the sample code from TRIGRID example2 feeded to SURFACE:
>
>> ; Create some random longitude points:
>>
>> lon = RANDOMU(seed, 50) * 360. - 180.
>>
>> ; Create some random latitude points:
>> lat = RANDOMU(seed, 50) * 180. - 90.
>>

```
>>  ; Make a fake function value to be passed to FVALUE. The system
>>  ; variable !DTOR contains the conversion value for degrees to
>>  ; radians.
>>  f = SIN(lon * !DTOR)^2 * COS(lat * !DTOR)
>>
>>  ; Perform a spherical triangulation:
>>  TRIANGULATE, lon, lat, tr, $
>>      SPHERE=s, FVALUE=f, /DEGREES
>>
>>  ; Perform a spherical triangulation using the values returned from
>>
>>  ; TRIANGULATE. The result, r, is a 180 by 91 element array:
>>  r=TRIGRID(f, SPHERE=s, [2.,2.],$
>>     [-180.,-90.,178.,90.], /DEGREES)
>>  surface, r
>>
>>  end
>>
>
>  Any cues welcome thanks
>
>  Sylvain Carette
>  VRML designer-composer
>
```

--
Ben Tupper
248 Lower Round Pond Road
POB 106
Bristol, ME 04539

Tel: (207) 563-1048
Email: PemaquidRiver@tidewater.net