I am sorry, I did not make clear that grid1 and grid2 are two
different irregular grids (lat1,lon1),(lat2,lon2). It is indeed
classical, I know, and I feel quite embarrassed!

So the code here:

```
 n1 = n_elements(ilon)
 triangulate, x1, y1, c ; Compute Delaunay triangulation
 out = GRIDDATA(x1,y1, LINDGEN(n1), XOUT=x2, YOUT=y2, /NEAREST_N,
TRIANGLES = c)
```

Is giving me, for each point in Grid2, the index of the closest point
in grid1. My question is, how to get the four closest points? The "bad
trick" that I used with GRIDDATA won't work here...

On Jul 8, 3:21 pm, David Fanning <n...@dfanning.com> wrote:
> Fabzi writes:
>> You probably have been asked many times, but once again this
>> apparently simple problem is driving me crazy. The problem is famous:
>
>> I have a 2D grid defined by x1 (dim 2 array, for example lons) and y1
>> (dim2 array, for example lats). And I want to fit it to a second grid
>> x2, y2. More precisely, I want to know the indexes in GRID1 that are
>> the closest to each of my points in GRID2. The output of my function
>> has then the same dimension as GRID2.
>
> Unless I completely miss the point, this seems to me
> to be a simple nearest neighbor interpolation from one
> grid to another. CONGRID has been used for this purpose
> for years. :-(
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.dfanning.com/

> Sepore ma de ni thui. ("Perhaps thou speakest truth.")