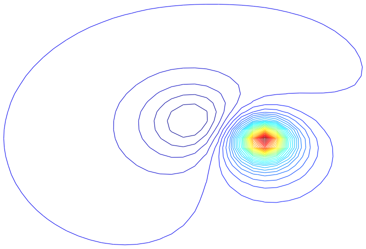
GEOL 481.3



POTENTIAL FIELD METHODS

LAB 4

GRIDDING, CONTOURING AND

REGIONAL-RESIDUAL SEPARATION by LEAST SQUARES

Gridding of data is always necessary before a contour map can be produced, because all contouring algorithms need to have the data on a regular, evenly spaced, grid. Data collected at irregular points in 2 dimensions, sometimes called XYZ data, needs to be interpolated to a regular grid. There are two mag data files called mag00.txt and reg00.txt on PAWS. This is the grid mag data from 2000. Despite the fact it appears to be on a regular grid, you still need to go through the gridding process. Also on PAWS is CompositFA.txt, a compilation of all the gravity data from field school 1996-present and saskgravr.txt a regional data set of various oil company data and field school data.

MATLAB has several utilities for gridding and contouring, and I have written a few more to make the process a little easier, and more specific to our needs. Suppose you have a file of irregularly spaced data in an array A, and that the first column of A is the x coordinates, the second column is the y coordinates, and the third column is the actual data (z coordinates). Then

```
[X,Y]=meshgrid([1:12]*10,[1:10]*2);
```

creates rectangular arrays of X and Y coordinates on a regular grid that is 10 rows by 12 columns. *meshgrid* is a built in MatLab m file.

Now you need to interpolate your irregularly spaced data onto this regular grid.

```
Z=griddata(A(:,1),A(:,2),A(:,3),X,Y);
```

The above line interpolates over the data in A, and returns values at the grid nodes defined by X and Y.

There are a number of different choices for the method of interpolation, usually speed is traded off against some other quality. With potential fields, it is most important to chose an interpolation scheme which produces a smooth interpolated surface, because potential fields are always smooth. *griddata* allows some choice of gridding method. If a sixth argument is added to the call of *griddata* as 'linear', 'cubic', 'nearest' or 'invdist', then the interpolation is linear, cubic, nearest neighbour, or inverse distance respectively. Cubic interpolation and inverse distance produce the smoothest results, so these are the best (of these) choices for potential fields data. There are many other interpolation schemes for data, some even better for potential fields data.

xyz2grid is an m file I wrote that includes calls to the above MatLab m files. *xyz2grid* automates the above procedure, ie it makes calls to *meshgrid* and *griddata* and you do not need to remember the details of these calls. The call is [X,Y,Z]=xyz2grid(x,y,z,'eg cubic'); where X,Y,Z are matrices of the gridded data and x,y,z are column vectors of the xyz data. Normally x would be easting etc. X and Y are the gridded x and y coordinates, and Z are the z values interpolated onto the X,Y grid.

When you call *xyz2grid* a map of station locations is first shown. This map will show you the overall plan of your data, including missing stations. Next you will be prompted for the dimensions of the grid you want to work on. You need to select these dimensions so that your grid is almost entirely contained by your samples. It should not be much larger in any dimension, because extrapolated data quickly loses any relevance and will produce Nan's (Not a Number). Your grid can of course be much smaller than the area of the xyz data. `You will also be offered to reduce the grid size if needed - select 'yes' or 'no'`

The m-file *meshgrid,* will not tolerate any station with two observations, so *xyz2grid* first calls *redundnt* to find repeat values. All repeats at a station are averaged to produce a single value. *redundnt* will also return the standard deviation of the repeats but if you have several repeats at one station it does not really do the calculation properly.

You will also be prompted for a sampling interval on the grid. This should not be much smaller than your sampling interval in the field (you cannot add detail by interpolating), nor can it be much larger, or you will run into aliasing problems. The field data were collected with a certain spacing to meet some objective, so it is best to have a similar spacing in the gridded data. Maximum, minimum and mean spacings in the field data are printed to screen to help you decide on an appropriate grid spacing, but sensibility is probably your best guide. Looking at the station location map will also help in selecting a spacing. Even if your big grid is exactly 10m by 10m, with no missing stations, you still need to run it through xyz2grid to put the station locations in the matrices X,Y,Z. If you select a 1m by 1m grid that is 50m by 50 m wide, then each of your X,Y,Z matrices are 50 by 50 points, and it may require considerable time to do the interpolation.

Now you can look at the gridded data with a variety of utilities. Try *contour(Z)*. The lower left corner is Z(1,1). You can specify the number of contour intervals as *contour(Z,20).* Notice that the x and y axes are just the number of rows and columns. You can give these the correct dimensions with *contour(X,Y,Z,20).* The above contour intervals are evenly spaced, if you want to customize the contour intervals use *contour(X,Y,Z,v),* where v is a vector that contains the contour levels you want. Finally, if you want contour intervals spaced at a particular interval, then try *contour(X,Y,Z,(max(max(Z))-min(min(Z)))/interval)*, where *interval* is the spacing you want.

If you want a three dimensional view use *surf(X,Y,Z).* If you want to overlay the actual data on this gridded map do this:

```
surf(X,Y,Z)
hold on
plot(A(:,1),A(:,2),'wo')
text(A(:,1),A(:,2),num2str(A(:,3))
```

You can change the viewing angle with *view(azimuth, elevation)* where azimuth and elevation (in deg) specify the viewpoint. The default is azimuth=-37.7, elevation= 30 (both in deg). Try a number of different views. *rotate3d* is also useful and lets you manipulate the view with the mouse. colormap is a utility that lets you adjust the color map of your image. Try *colormap(hot), colormap(cool), colormap(gray), colormap(pink), colormap(copper), colormap('default')* will get you back to something sensible. *colormap(hot(n)),* where n is an integer, lets you adjust the number of levels of color. You can also try *brighten(x)* where x is [-1 1] if you want to brighten or darken the shading. If you want to get rid of the mesh lines try *shading interp.*
title, xlabel and ylabel, will add labels to the graph and the X and Y axes. Type help title etc for a description.

Another utility is *mesh(X,Y,Z)* which just draws the net, with colored strings, instead of coloring the whole map. This is called a curtain plot, or fishnet plot. *meshc(X,Y,Z),* will draw a contour map under the curtain map.
ZMASK=gridmask(X,Y,Z,'post') is another utility you may find useful. The 'post,' if present, shows the positions of the original grid. *gridmask* allows you to mask out

selected parts of your gridded data. For example, if you have one station which is suspect then you can use gridmask to mask this station. Also, if the dynamic range of your observations is large, then all of the available contour lines are used to display this, and subtle features will not be apparent. You select the area to mask by clicking on the bottom left and top right corners of a rectangular area you want masked. *gridmask* will continue prompting you for blocks of data to mask, until you tell it to stop. The unmasked data can be contoured with *contour(X,Y,ZMASK).*

1) Contour your mag data at a suitable station spacing. Remember that there was a large constant value subtracted from all measurements so you have numbers much smaller than the total field. Mask out the large anomaly with *gridmask* and recontour to see if there are any smaller anomalies that you might want to look at in addition to the large anomaly. If you took good notes in the field you should be able to correlate many of these small anomalies with observed features.

2) Contour the saskgravr Bouguer data (the file is free air so you have to calculate Bouguer.) The area covered by this map is much larger than our survey area, but the extra points allow interpolation to the edges of our data.

3) Isolate the patch of data around where you collected data and draw a contour map. The map will look terrible because we have a poor distribution of points E-W, but you can get a sense of the size of the gravel deposit.

To least squares fit a plane to gridded data do the following

```
[n,m]=size(Z);
nm=n*m;
A1=[ones(nm,1) reshape(X,nm,1) reshape(Y,nm,1)];
Z1=reshape(Z,nm,1);
```

In lab report, explain what this means.

This creates two vectors A1 and Z1. A1 has three columns and the same number of rows as there are elements of Z, Z1 is one column with the same number of elements as Z. *reshape* has reshaped Z into a column vector, Z1.

```
plane=A1\ Z1
```

This last line gives you the coefficients of the plane. The \ is *matlab* shorthand for divide the vector $Z1$ by the matrix $A1$ (provided this is dimensionally correct) and this division is the least squares solution for the coefficients of the plane. Now construct a column vector with the actual values of the plane.

```
P1=A1*plane;
```

P1 is now a column vector that describes the plane. To turn this into an array with the same dimensions as Z:

P=reshape(P1,n,m);

P is now an array with the same dimensions as Z, and it is the gridded plane that was least squares fit to Z.

Normally when you do a detailed survey you would sample some stations well outside your grid, so that you could use these to define a regional over your grid. In your case you have the 10m by 10 m grid for this purpose.

Polynomial fitting has been automated in an m-file called *polyregional.m*. The call is *[ZREG,Zreg] = polyregional(XBIG,YBIG,ZBIG,X,Y,Z); polyregional* fits a polynomial surface to data on a big grid defined by XBIG, YBIG, ZBIG, and projects it onto a small grid X,Y,Z contained within the big grid. XBIG,YBIG,ZBIG,X,Y,Z are matrices created by *xyz2grid.* X,Y,Z do not have to be gridded on the same spacing as XBIG,YBIG,ZBIG, but the boundaries of X,Y should be entirely within the boundaries of XBIG,YBIG. the polynomial surface defined on XBIG,YBIG is returned as ZREG, and the polynomial surface defined on X,Y is returned as Zreg. XBIG,YBIG,ZBIG might be identical to X,Y,Z if your have no big grid to work with.

Again, you do not want anomalous regions, or points, to influence the choice of regional, so you can mask these out (*polyregional* calls *gridmask* for this purpose). You will be prompted for a choice of polynomial - linear, quadratic, cubic, or quartic, are the only choices available.

4) Try *polyregional* with different polynomials, and different maskings. Normally you should compare your regional with published maps, but in the case of our mag survey the local area is too small.

polyregional has some limitations as a regional residual tool. An improvement would be to replace the grids XBIG,YBIG,ZBIG, with the raw xyz data, fit a polynomial to these data and then project onto the detailed grid. This scheme would not allow you to mask out certain areas. As it is, *polyregional* calculates the regional based on interpolated data, not the raw observations. This will not matter much for your data because the XBIG,YBIG,ZBIG are nearly identical to the raw data, but it could be a problem for other data sets. For now, the windowing advantage of using XBIG etc, outweighs the disadvantage.

5) Saskgravr is field school data saskgrav data combined. perform a regional residual separation on the field school data and create a contour map.