CIVL 8126 / CERI - 8104
Analyzing Spatial Data

Reading: "MATLAB Recipes for Earth Sciences", Trauth Chapter 7 (Spatial Data)

Submit a short report with figures documenting your result! You will also have to submit your MATLAB code with comments.

Required files for this assignment: `normalfault.txt.`
Required software: **rot_Mz.m and points2angle.m** functions from Canvas.

This assignment will focus on the analysis of spatial data in cartesian coordinates. Spatial data can come in many shapes and forms. Data in cartesian coordinates are probably the easiest to deal with. In a later homework, we will analyze earthquake catalogs which are usually reported in a spherical or polar coordinate system with a complicated (UTC) time stamp. For now, we will process data with coordinates in X,Y,Z.

The assignment has the objective to:

1. Develop a detailed understanding of file structure and data tables.
2. Learn how to use basic matrix operations
3. Produce high-quality images of data in a cartesian coordinate system
4. Calculate slopes and angles from 3D data

Stratigraphic Layer Topography

In the first part, you will analyze one specific stratigraphic surface that has been displaced by a normal fault. The data file (`normalfault.txt`) provides different measurements of the location of this surface in X, Y and Z coordinates. The picture above is a cross-section of a normal fault. To better understand what the data file is imaging, you can trace the upper surface of the darkest red layer in the picture above and extrapolate that layer in 3-D. At the end, of this section, you will compute the dip angle and dip direction of this specific normal fault.

1. **Load data, interpolate measurements** – Create a MATLAB script called: Hw2_1.m
    a. Download the following data file from Canvas: `normalfault.txt.` (Hint: check the `>>load` command in MATLAB. Either, specify the absolute location of the file or copy is to your cwd.)
    For the first part, you will have to only analyze the data in `normalfault.txt,` we will get to the second file later.
    b. Inspect the data and find the length of each column as well as the min. and max. values. Create three pairs of parameters (xmin, xmax, ymin, ymax, zmin, zmax.) that encapsulate the range of observed values. We will later use these parameters to create data grids and set the limits of our plots.
    c. Plot the raw data in 2D with text labels for the z-components. Use the commands:
    >>`plot` and
    >>`text`
    Use `text()` to plot the x and y coordinates of each data point and add a label to each point that shows the corresponding z value. To create these labels, you will have to convert the floating point numbers to strings using
    >>`num2str`
    Include this figure in your report together with a figure caption!

We will now interpolate the data to fill a regular-spaced grid using a spline interpolation technique.

    d. Create two coordinate vectors from your xmin, xmax, and ymin, ymax parameters. Use
    >>`meshgrid` to create the coordinate matrices.
    e. Now you have to interpolate the measured Z values across this regular grid. Type:
    >>`help griddata` in the matlab command line prompt to find out how to do that.
    f. Plot the interpolated surface using the following sets of commands:

```
a_con = zmin:10:zmax;%specify contour line spacing
pcolor(XX, YY, ZZ), shading flat, hold on;
cbar = colorbar;
cbar.Label.String = 'Z (m)';
[c,h] = contour(XX,YY,ZZ, a_con, 'k');
clabel( c, h)
```

    g. You can create a third plot of a 3D meshed surface
    >>`mesh( XX, YY, ZZ),`
    and a 3D colored surface
    >>`surfc`(XX,YY, ZZ). Explore how to juxtapose these different plots using the
    >>`subplot`

Command!
Add labels to each axis as well as color bars where needed and submit these figures with your report.

2. Create a separate MATLAB script to **compute the average dip angle** of the fault
    a. For this purpose, you should demean and rotate the original data matrix XYZ, so that the fault strike is roughly aligned with the y-axis. Use the function **rot_Mz.m** to rotate the 3D surface about the z-axis.

    b. Interpolate and plot the rotated data points.

    c. Now take a look at individual profiles across the fault (i.e. along the y-direction which are the rows of the rotated ZZ matrix). Plot both the individual profiles and the final average of all profiles.

    d. Use `ginput(2)` to specify two points along the normal fault and compute the fault dip angle by using: `atan2(y(2)-y(1),x(2)-x(1)),` where atan() is the arc tangent between the two points x and y. You can use the function: **points2angle.m** to compute the angle between two points in cartesian coordinates.

    e. Bonus: What is the dip angle and how does that compare with your expectation based on Anderson's theory of faulting?