THE UNIVERSITY OF MEMPHIS

CENTER FOR EARTHQUAKE RESEARCH AND INFORMATION

Activity 5

"Spatial Data - Normal fault Orientation and Dip"

DATA ANALYSIS IN GEOPHYSICS

CERI 8104

FALL SEMESTER - 2025

Presented by:

Adonay Martinez-Coto

September 28, 2025

PART I - LOADING AND INTERPOLATING DATA

INSTRUCTIONS:

1. Download the data file 'normalfault.txt' from Canvas. 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. 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.
2. Create two coordinate vectors from your xmin, xmax, and ymin, ymax parameters. Use '»meshgrid' to create the coordinate matrices.
3. 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.
4. Plot the interpolated surface using.
5. Create a third plot of a 3D meshed surface and a 3D colored surface, put labels to each axis as well as color bars where needed.

RESULTS:

After downloaded the data, the data has three columns, the first column is the X coordinate, the second column is the Y coordinate and the third column is the Z (elevation) coordinate. The length of each column is 79 rows and the min and max values for each column are:

- X range: 421.22 to 469.25 Y range: 70.69 to 118.20 For Topography (Z): -27.00 to 21.00

The following figures show the results of the plots:

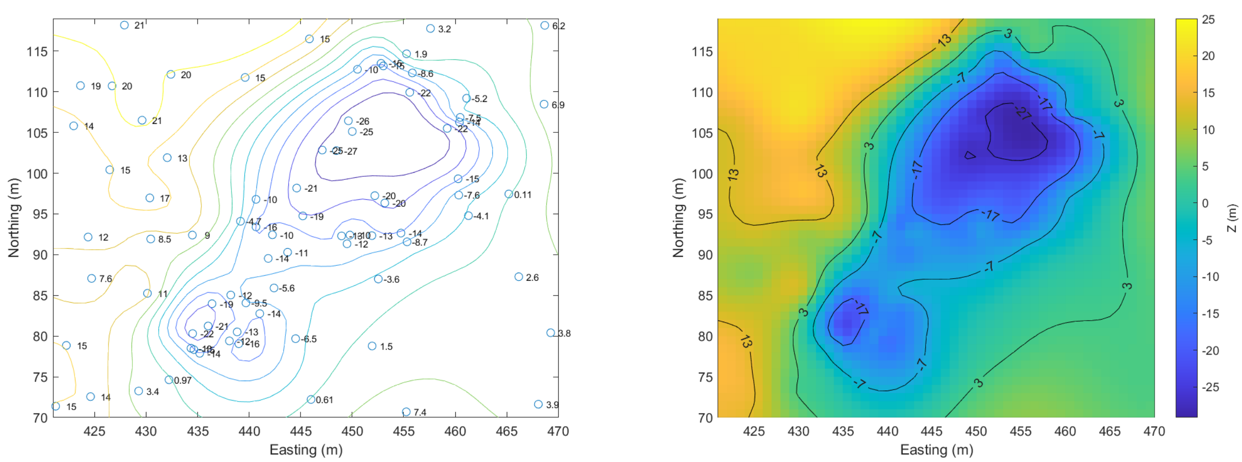

Figure 1 – *Normal Fault Data. Left: Raw data with Z labels. Right: Interpolated contour map of the data.*

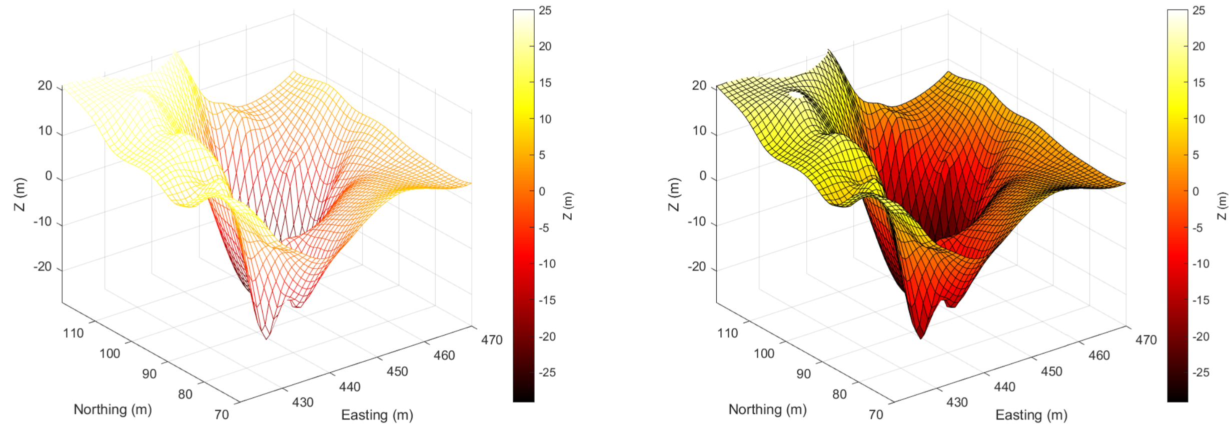

Figure 2 – *Normal Fault Data. Left: 3D mesh for the data. Right: 3D colored surface for the data.*

Part II - Compute the average Dip Angle

Instructions:

1. Demean and rotate the original data matrix XYZ, so that the fault strike is roughly aligned with the y-axis. Use the function `rotz` to rotate the 3D surface about the z-axis.
2. Interpolate and plot the rotated data points.
3. 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.
4. 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: `atan2` to compute the angle between two points in cartesian coordinates.
5. Bonus: What is the dip angle and how does that compare with your expectation based on Anderson's theory of faulting?

Results:

The preprocesing steps that include the demeaning and rotation of the data points are where done in the data, and the results are shown in following figure.

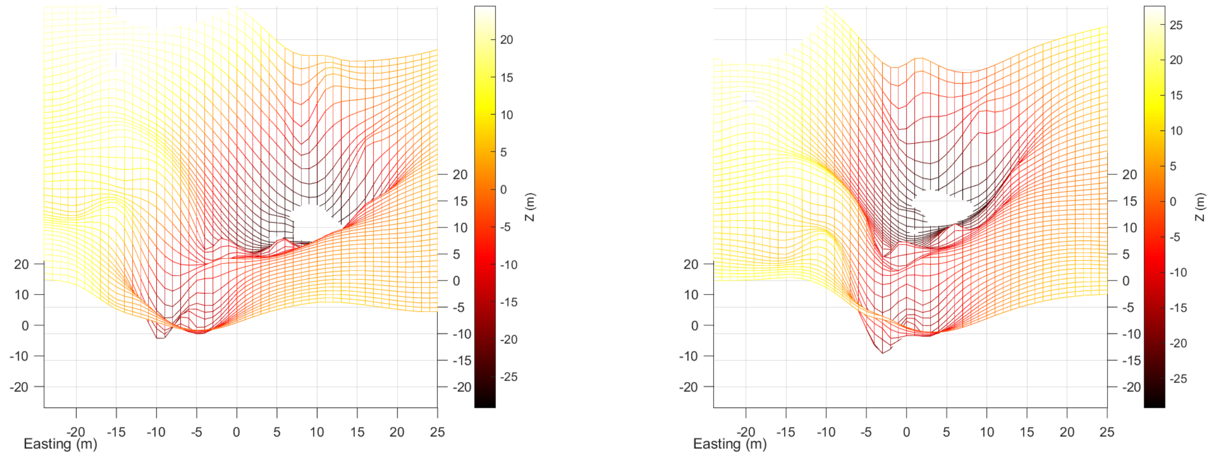

Figure 3 – *3D surface of the topography data before (left) and after (right) rotation.*

The individual profiles along the y-direction (rows of the rotated ZZ matrix) and the average profile are shown in the following figure.

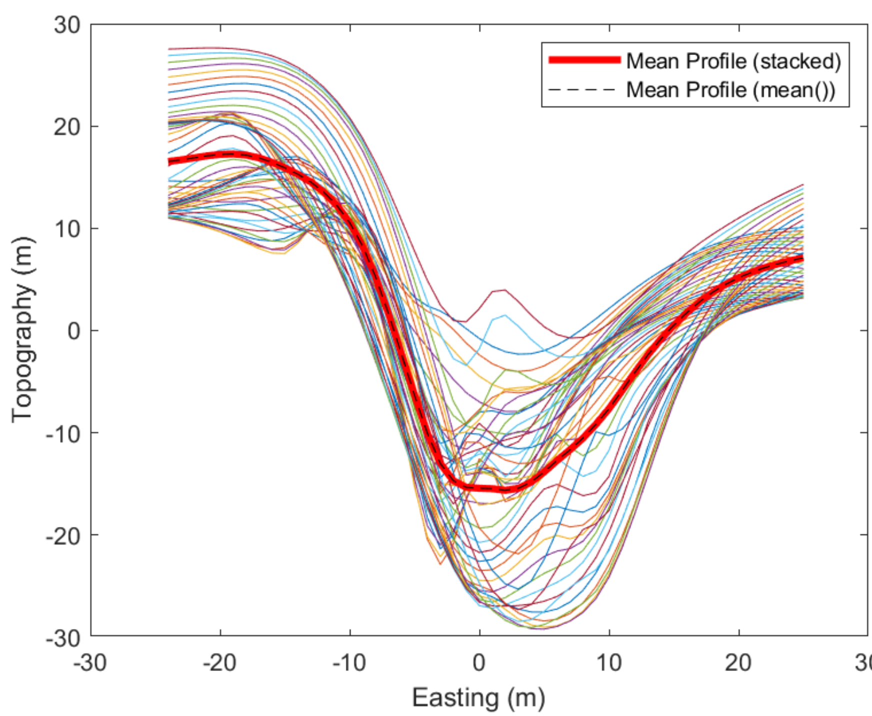

Figure 4 – *Individual profiles along the y-direction (rows of the rotated ZZ matrix) and the average profile (red line).*

With figure 4 the dip angle was estimated selecting two points as indicated, the resulting dip angle after several attempts was approximately 73 degrees. The Anderson's theory of faulting propose a model that relates the state of stress to the orientation and type of faults that may form in the Earth's crust, where the one principal stress direction is assumed to be vertical. For normal faulting, where the maximum principal stress is vertical, Anderson's theory predicts an optimal dip angle of approximately $\tilde{6}0$ degrees, assuming standard rock friction values [Célérier, 2008].

This theoretical prediction is based on the balance of forces acting on a fault plane, where the dip angle is influenced by the internal friction of the rock. However, empirical observations often show that actual fault dips can vary significantly from this theoretical value due to factors such as rock heterogeneity, pre-existing weaknesses, and local stress variations [Célérier, 2008].

For our case in this simplified analysis, where the dip angle is approximately calculated, showed an steeper value compared to the theoretical proposed value of around 60 degrees. But this can be attributed to the limitation of the way we are calculating the dip angle in our code, as well as the quality of the data and the specific geological context of the fault being analyzed. But this does not mean that our fault is not a normal fault, as in the nature we have a wide range of geological settings, also should be noted that Anderson's theory is a simplification that assumes completely vertical maximum stress and uniform rock properties, which may not hold true in all real-world scenarios.

Part III - Creating an Animation

Instructions:

Create a simple .gif animation of the rotating 3D surface.

Results:

For this part I decided to deviate a little bit from the original instructions and create both a .gif and a .mp4 video of the rotating 3D surface. But instead of using ImageMagick to create the animation, I used MATLAB's built-in functions to create both the .gif and .mp4 video.
The code used to create the animations is included in the attached MATLAB scripts. But here the brief explanation of how the animations were created is:

1. A loop was created to iterate through the rotation angles from 0 to 360 degrees in steps of 10 degrees (I use 10 degrees to have more frames in the animation).
2. For each rotation angle, the 3D surface was rotated using the 'rot_Mz', as indicated in the previous parts.
3. The rotated surface was then plotted using the 'surfc' function.
4. The current figure was captured as a frame using the 'getframe' function.
5. The frame was converted to an RGB image using the 'frame2im' function.
6. The RGB image was then converted to an indexed image using the 'rgb2ind' function, which is necessary for creating a .gif file, since .gif files are saved as indexed images with a color map.
7. The indexed image was then written to the .gif file using the 'imwrite' function. For the first frame, the 'Loopcount' parameter was set to 'inf' to create an infinite loop, and for subsequent frames, the 'WriteMode' parameter was set to 'append' to add frames to the existing file.
8. The frame was also written to the .mp4 video file using the 'writeVideo' function.

The resulting animations can be found in the attached files: ‘fault_rotate.gif‘ and ‘fault_rotate.mp4‘. Or in the links below:

- fault_rotate.gif
- fault_rotate.mp4

References

Célérier, B. (2008). Seeking Anderson’s faulting in seismicity: A centennial celebration. *Reviews of Geophysics*, 46(4).