Computational Project 1

Due: 01/29/2015, noon

Create a single function called cp1_MEID (no dash) with no inputs or outputs (just leave those empty). Publish as a pdf and print this function to turn in a hard copy in class. Also submit your matlab file to the dropbox on D2L. Please label parts of your code with the corresponding section number below and make sure to comment throughout.

First make a 3D map of Washington state topology:

1. Load the database WA topo.txt of vertical altitudes [m] for Washington in a latitude × longitude grid into a matrix using the load('WA_topo.txt') command (make sure this file is in your current directory). Use the size command to figure out how the data is arranged in your matrix in terms of the number of rows and columns.

2. Topo maps that you see often de-emphasize rugged vertical variations by plotting the square root of altitude. To do this we need to first get rid of any negative altitudes (i.e., oceans) in our topo matrix. So replace all negative altitudes with zero. Convert all positive values from units of m to ft.

3. Make a plot of the square root of the topo data using the surf command. The plot may look black at first, so figure out how to set the EdgeColor property to none so that the grid-box edges are not displayed. Make sure you haven't plotted the state upside-down, backwards, etc. Label the axis, and add a figure title. Experiment with setting the view to a pair of angles such that the map is easily recognizable and the axis labels legible. Also, the plot will by default end up square. Adjust the x aspect ratio using the pbaspect command so that Washington is accurately proportioned (it is about 240 miles tall and 360 miles wide).

4. The tallest points in the western part of the state are all on Mt. Rainier. Less obvious are the locations of the highest peaks in Eastern

Washington. Add code that finds the height, matrix row index, and matrix column index of the 3 tallest peaks in Eastern Washington. The eastern part of the state corresponds to columns 500 through 940.

5. Using the hold command and the plot3 command, add to your plot black upward pointing triangles on top of the three tallest peaks in Eastern Washington.

6. Write a separate function that takes as input arguments a latitude and longitude and returns as output the corresponding row and column indices of the topography data matrix. Use the following facts: 1

- The latitude range of the data is 45.56666700000° to 49.00000000000°.
- The longitude range of the data is -124.71666700000° to -116.88333300000°.
- Each grid cell is a square with sides of length d = 0.00833333334°.
- The floor command can be used to round a decimal number down to the closest lower integer. Have the code print out an error message and return an index pair of -999,-999 if the requested latitude and longitude lie outside of Washington. To help you, here is the formula for determining the index corresponding to a particular longitude:

index = floor((longitude – western longitude edge)/(grid cell width)) + 1

7. Use your function from part 6 and the plot3 command, add to your program some code that draws a filled black circle on your map, at the correct altitude, at the following city coordinates:

- Seattle (47.6046°,-122.33057°)
- Spokane (47.6589°,-117.4250°)
- Wenatchee (47.4233°,-120.3253°)

II. Now write a separate function to analyze vertical variation for some possible bike rides across the state. For simplicity, we will consider rides that only head straight from south to north, regardless of roads or terrain (see part 3). Your function should take as input a pair of latitude and longitude coordinates that specify the starting (lat_1, lon_1) and ending (lat_2, lon_2) points of the ride, and the topo matrix. It should perform the calculations and actions described in items 1 - 14.

8. First extract a vector z of n altitudes values from the topo matrix between the input pair of coordinates such that z_1 is the altitude at (lat_1, lon_1) and z_n is the altitude at (lat_2, lon_2). You should use your function from part 6 to determine the indices of the topo matrix that define the starting and ending position of z. An example of extracting a vector from a matrix is the following code, which displays only columns 3 through 5 of row 2 of a 10 × 10 matrix

```
A=magic(10)

vector = A(2,3:5)
```

9. Calculate the maximum altitude along the ride. Display the calculated value in the command window.

10. Calculate the average altitude along the ride:

$$\bar{z} = \frac{1}{n}\sum_{i}^{n} z_i$$

Display the calculated value in the command window.

11. Calculate the running average altitude at each point i along the ride using the following iterative formula

$$\bar{a}_{i+1} = \bar{a}_i + \frac{1}{i+1}\left(z_{i+1} - \bar{a}_i\right)$$

where $a_1 = z_1$

12. Calculate the latitude x_i at each point i along the ride.

13. Subdivide a new plotting window into two panels using the subplot command. Plot the altitude vector z as a function of latitude x using a red line. Set the axis bounds to tightly capture the plot range. Label the axis.

14. In the second panel, plot the running average elevation $\bar{a}$ as a function of latitude x as a dotted blue line.

III. Add to the bottom of your code from part I some new code that will call your function from Part II for the inputs listed below. Use the figure command to open a new plot window prior to each call to your function.

15. Seattle to Canada: (47.6046°,-122.3305°) to (49.00°,-122.3305°)

16. Walla Walla to Spokane: (46.0650°,-117.4250°) to (47.6589°,-117.4250°).

17. Wenatchee to wilderness: (47.4233°,-120.3253°) to (48.800°,-120.3253°).