ENGR 105 – Introduction to Scientific Computing

Assignment 9

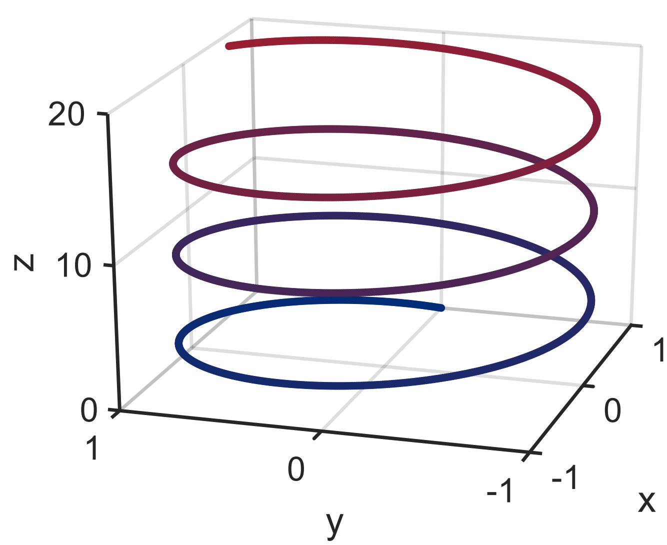
Due by 11:59 pm on Weds. 3/23/2022 on Gradescope

Problem 1 (20 pts): A helix can be constructed in three dimensions using the following parametric equations.

$$x = \cos(t)$$
$$y = \sin(t)$$
$$z = t$$

Create a script that produces a pleasing, three-dimensional plot of the curve described by *x*, *y*, and *z* over the domain $0 \le t \le 20$ in three dimensions. When viewed reasonably by eye, the curve should look smooth and vary smoothly from one custom color to a second custom color. All color specifications should be driven by RGB triplets. The plot should include a perspective view projection, labels on all axes, larger than default axis label and tick label font sizes, and a grid. The plot should appear three dimensional in its presentation and all plot modifications should be achieved through code.

Your plot might look something like the following, which varies from Penn blue to red.



Upload the script that produces the plot and a .jpg copy of the resulting plot. Identify the names of all files associated with this problem in your README.txt.

Problem 2 (35 pts): You have obtained a summer research position with Apple. The company is currently investigating reliability issues of the electrical connectors on their lightning cables. In particular, a new, low-cost manufacturing process has led to high resistances between the lightning plugs and the electrical connections with which they mate.

You are scanning these metal surfaces using atomic force microscopy[1] (AFM) with the hope of identifying variations in the surface of the electrical connecter that may contribute to these reliability issues. The measurement consists of rastering the surface with a nanoscale probe tip while measuring the topography of the surface, the friction force between the probe tip and surface, and current through the probe tip-surface junction. The resulting data sets are output as two-dimensional matrices of data representing the magnitude of a response at regularly-spaced locations across the sample surface.

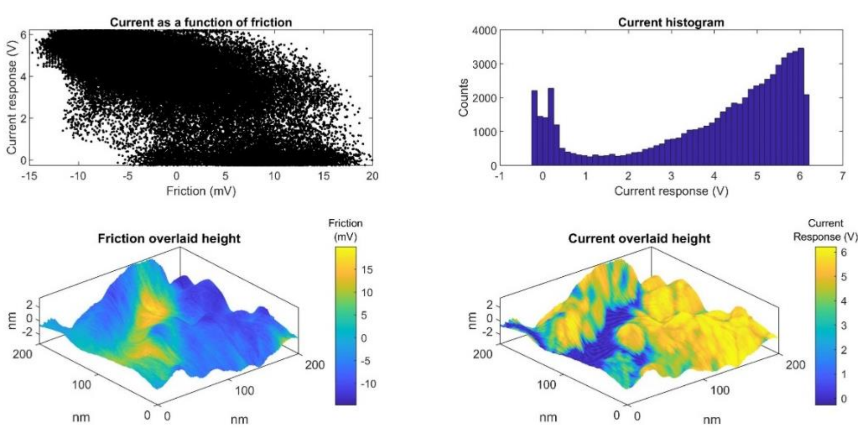
You have been performing a lot of these AFM scans and would like to visually summarize your findings for each data file. You decide to use MATLAB to quickly produce standardized plots of each data set and save the plots to your working folder. To do this, you concoct a plotting function with the following function declaration.

```
function plotAFMdata(H,F,I,ss,saveName)
```

where `H` is a 2D matrix of height data, `F` is a 2D matrix of friction data, `I` is a 2D matrix of current response data, `ss` is a scalar describing the lateral scan size of the data set in nanometers, and `saveName` is a character vector that represents the data file name to which the generated figure will be saved. You can assume that `H`, `F`, and `I` will always be square matrices of some size n x n. Of note, the height, friction, and current data set with which you are provided contain 256 x 256 surface map points, which correspond to a lateral scan size, `ss`, of 200 nm along each edge.

The data is provided to you from the AFM as a .mat file with fields corresponding to `'height'`, `'friction'`, and `'current'` (current response) in units of nanometers, millivolts, and volts, respectively. You have been provided with the data set `AFMdata0001.mat` to test your function.

`plotAFMdata` should produce a figure closely resembling the following figure.



[1] See http://en.wikipedia.org/wiki/Atomic_force_microscopy for more information about atomic force microscopy.

`plotAFMdata` should produce a single figure with the following axes and specifications.

Southwest and southeast axes: Produce a three dimensional plot of height with color provided by the friction data and a three dimensional plot of height with color provided by the current response data. The plots should include the following modifications.

- Each axis should extend no further than the domain, range, and height of the data.
- The azimuth and elevation should be specified.
- Do not include a grid.
- Include a bounding box.
- Include a color bar.
- Include a title on the color bar.

Northwest axes: Plot current response as a function of friction.

- You can let MATLAB automatically select the domain and range.

Northeast axes: Produce a histogram of the current response for all current data points.

- Specify a domain of -1 to 7 but do not specify the range.

All plot modifications occur via code. All plots should include the following elements, specifications, and modifications.

- Include labels on all axes.
- Include titles on all plots.
- Specify the font size and font name (Arial) for all text elements.
- Specify a white figure background color.

Use `axes()` to specify the position of each plot instead of `subplot()`.

You want to ensure that your code will display the plot correctly on any modern computer. Therefore, your figure should be positioned in the center of the computer screen and have dimensions of 1000 x 460 pixels (width x height).

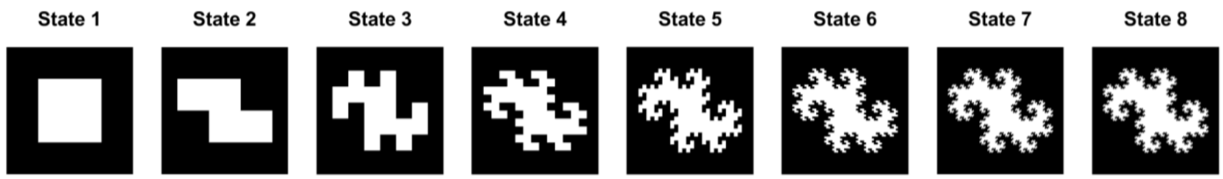
`plotAFMdata` should save the figure to the working folder under the name specified by `saveName` as a 24-bit jpeg with a resolution of 300 pixels/inch. `plotAFMdata` should then close the figure window after saving the jpeg.

Create an associated script to test your function that loads the test data and launches the `plotAFMdata` function.

Upload the `plotAFMdata` function, the test script, and a .jpg copy of the resulting test plot. Identify the names of all files associated with this problem in your README.txt.

Problem 3 (45 pts): Fractals frequently appear in nature. The structure of the lung, diffusion, Romanesco broccoli[2], and surface roughness[3] all display fractal qualities.

A fractal-like pattern can be generated by shearing a square by increasingly smaller divisions and displacements in alternating horizontal and vertical directions. This is shown in the panel of images below and in the reference video `shear_square.avi`. The white square in the starting state (state 1) has an edge length half that of the axes.



As well, you may find `shear_square_advanced.avi` informative as you consider the patterns and algorithms that inform your approach to creating an animation that replicates `shear_square.avi`.

Create a script and/or function(s) that recreates `shear_square.avi` under the name `my_shear_square.avi` or `my_shear_square.mp4`.

Animations in MATLAB can be achieved using code like the following. In this example, a 20 x 20 matrix of random numbers is plotted once and then the color data of the plot is updated as the information in the matrix is updated.

```
% Generate a 20 x 20 matrix of random numbers.
M = rand(20);

% Plot the matrix using surface(). Remove border lines between
% the pixels, specify a gray color map, and remove axis
% labels/ticks.
s = surface(M,'edgecolor', 'none');
colormap gray, axis off

% Pause for 1 second.
pause(1)

% Update the matrix and the plot, pausing between each update.
for jj = 1:10
    M = rand(20);
    s.CData = M;
    pause(1)
end
```

---

[2] http://en.wikipedia.org/wiki/Romanesco_broccoli

[3] B. N. J. Persson, "Theory of rubber friction and contact mechanics," *Journal of Chemical Physics*, vol. 115, no. 8, pp. 3840–3861, 2001.

As a bonus challenge (for bonus points), you may wish to create a more detailed version of the pattern generator that animates displacements during the shearing process and/or works with a wider palette of colors like that of `shear_square_advanced.avi`.

Upload your scripts and/or function(s) and video file(s). Identify how your code should be launched and identify the names of all files associated with this problem in your README.txt.