Homework 14: Surface Plotting & Animation

Project Homework

Instructions:

This homework is different than the homework assignments that you have done before. Instead of having many problems to solve with different functions, you will be graded only on one function (Problem 5, `flyPropellerHat`), which will use bodies of rotation, surface plotting, and rotations to animate a scene of a propeller hat flying around.

Instead, problems 1 through 4 are designed to assist you in creating separate components (e.g. a cap, a propeller) that will be used together in the `flyPropellerHat` animation function. These problems are not required in order to receive full credit on this homework. However, it is *highly recommended* that you utilize them in your `flyPropellerHat` function. They will be helpful in breaking down the final animation into manageable components, and they will also help organize your code. In addition, if you correctly complete any of problems 1 through 4, you will receive partial credit even if your `flyPropellerHat` function does not work correctly.

Notes:

You have a lot of freedom in this project to change things up and be creative. That being said, you can change the inputs and outputs of the helper functions provided as the first few drill problems to create what you need for the final animation. Further, you can create additional functions if you want to do more than what the provided function guidelines offer.

Grading Scheme:

Unlike previous homework assignments, this homework will not be graded with the auto grader. Instead, the `flyPropellerHat` function will be hand-graded by your section TAs using the rubric provided below. Seventy percent (70%) of your grade for this homework will come from this hand-graded score.

The remaining thirty percent (30%) of your grade will be based on your ability to demonstrate and explain how your `flyPropellerHat` function works to your section TA during a demo. This demo will be done in recitation during the week before finals (commonly referred to as Dead Week). The demo gives you a chance to boost your homework score, even if there were some errors in your code when it was submitted. If you are able to knowledgeably answer your TA's questions about how your code works or explain any errors in your code, you will receive full credit on the demo.

Make sure you look through the rubric provided to ensure you have fulfilled all the requirements for this project homework.

Extra Credit:

The animation aspect of this homework will give you an opportunity to be creative. If you go above and beyond what the problem statement specifies, there is potential to receive extra credit. Extra credit will be awarded by your TA based on how many creative additions you make to the animation. You **must** include a commented statement at the bottom of your `flyPropellerHat` file (below all of the code) detailing the additions that you made *or that you did not make any additions*. ***The TAs are NOT responsible for helping you implement your extra credit!***

Example Extra Credit Additions:

- Create more background stationary/moving objects than required
- Add a background
- Put clouds in the animation
- Make the hat motion non-circular

Disclaimer:

Not all of the material on this homework will be functions we teach you in class. Most of MATLAB is like that. You have all the basic skills required to do this, but use the Internet/Textbook/MATLAB help resources to figure out the specifics. This assignment gives you the opportunity to go above and beyond, read the documentation on certain functions and functionality, and have fun with it!

Good Luck!

Project Homework: Problem 1

Function Name: makeCap

Inputs:

1. (*double*) A radius for the cap
2. (*double*) A vector of coefficients for a polynomial

Outputs:

1. (*double*) An MxM array of xx-data
2. (*double*) An MxM array of yy-data
3. (*double*) An MxM array of zz-data

Function Description:

Write a function called `makeCap` that takes in a radius for the propeller hat's cap and a vector of coefficients for a polynomial, and then creates the body of the propeller hat. To do this, use bodies of rotation to make the three dimensional coordinates for the cap. This function will create xx, yy, and zz mesh data arrays as if the input polynomial were rotated around the z-axis. This cap may be used for plotting the cap of the propeller hat.

Notes:

– You may find the `meshgrid()` and `linspace()` functions useful.

Hints:

– You may find a fifth degree polynomial to be a good fit for the propeller hat's cap.

Project Homework: Problem 2

Function Name: makeBill

Inputs:

1. (*double*) A length for the bill
2. (*double*) A radius for the bill

Outputs:

1. (*double*) An MxM array of xx-data
2. (*double*) An MxM array of yy-data
3. (*double*) An MxM array of zz-data

Function Description:

Write a function called `makeBill` that takes in a length for a half cylinder and a radius for that half cylinder, and rotates around the x axis to create a half-cylinder shape. This function should rotate a line around the x axis to create xx, yy, and zz data. You should use bodies of rotation to accomplish this. You may NOT use the `cylinder()` function on this problem.

Notes:

- You may find the `meshgrid()` and `linspace()` functions useful.

Project Homework: Problem 3

Function Name: makePropeller

Inputs:
1. (*double*) A minimum x point
2. (*double*) A maximum x point
3. (*double*) A radius for the propeller

Outputs:
1. (*double*) An MxM array of xx-data
2. (*double*) An MxM array of yy-data
3. (*double*) An MxM array of zz-data

Function Description:

Write a function called `makePropeller` that takes in a minimum and maximum location for x data as well as a radius. This function will rotate a sin(x) function around the x axis. The x range of the sine wave should be within the minimum and maximum x locations input, and the sine wave will be amplified by the given input radius. To do this, use bodies of rotation to make three dimensional, xx, yy, and zz coordinate, mesh data arrays that may be used for plotting the propeller of the propeller hat.

Using a sine wave is a suggested method to create the propeller for the propeller hat. You are allowed to get creative with this one! Depending on exactly how you want your propeller to look, you may want to change this function.

Notes:
- You may find the `meshgrid()` and `linspace()` functions useful.

Project Homework: Problem 4

Function Name: rotateObject

Inputs:

1. (*double*) An MxM array of xx data
2. (*double*) An MxM array of yy data
3. (*double*) An MxM array of zz data
4. (*char*) Axis to rotate around
5. (*double*) An angle through which to rotate the data

Outputs:

1. (*double*) An MxM array of new xx data
2. (*double*) An MxM array of new yy data
3. (*double*) An MxM array of new zz data

Function Description:

Write a function called `rotateObject` that takes in xx, yy, and zz mesh arrays for an object, as well as an angle theta through which to rotate the object, and outputs the object's new xx, yy, and zz mesh data. You will want use rotation matrices to rotate your object.

Rotating coordinates in 3D is very similar to rotating in two dimensions; you just have a different rotation matrix. The counterclockwise rotation matrix for the third dimensions around the z-axis is as follows:

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

For more information on rotation matrices, and for the 3D rotation matrices around the x and y axes, you may visit: https://en.wikipedia.org/wiki/Rotation_matrix.

Notes:

- You may **not** use the `rotate()` function for this problem.
- A shape must be displaced from the origin to see the result of a rotation.
- Recall that the order that you multiply your data and the matrix will determine if your object rotates clockwise or counterclockwise.

Function Name: flyPropellerHat

Inputs:

You are not required to have any inputs to this function. For creativity's sake, we have provided possible functions you could implement in your code. If your code requires inputs, make sure you include commented instructions on how to use those inputs and an example input for the function below where you list your extra credit.

Outputs:

(*none*)

Plot Outputs:

1. Animated plot of flying propeller hat

Function Description:

This is the function you will be graded on overall for this homework assignment. Now that you have written functions to create pieces of a propeller hat and to rotate those pieces in a rotational motion, write a MATLAB program that will produce an animated scene of a propeller hat flying.

Because creating an animation to incorporate your other functions is challenging, skeleton code has been provided for you inside `flyPropellerHat.m`. This skeleton code contains comments that should guide you in using your other functions from this homework to create the animation.

The file `flyPropellerHat_sample.p` has also been provided for you to see an example of what your `flyPropellerHat` animation may look like.

Requirements for `flyPropellerHat`:

- You must have a propeller hat with a cap, bill, and propeller.
- Your propeller hat must be rotating.
- Your propeller must be rotating independently of the propeller hat's motion.
- You must have one object that is not rotating.
- The hat must be a solid object.
- The hat cannot be the default color.
- The hat must be visible on the screen throughout the entire animation.

- Axes are turned off.
- Comments are at the bottom of the file that outline creative changes made or a statement that no creative changes were made.
- Directions are included on how to run your function if you add inputs to the function.