

COMS W3101-2 – Spring 2010

Homework 2

The main goal of the first homework is to have you practice with plotting (2D and 3D), use control flow and implement efficient code in MATLAB.

For this assignment you should write a single m-script file called hw2.m. At the top of the file you should put your name and ID in a comment. There are 5 exercises in this assignment, each worth 20 points, for a total of 100 points. For each exercise you will have to write a single or series of MATLAB commands. Make sure you place a comment before the code solving each of the exercises, in order to specify which part of code solves what exercise and to write the answers to the exercise.

This homework is due on March 23, at beginning of class, no exceptions. Put your code (.m file) and any additional files in a single folder, name it youruni_hw_2 and zip it. Upload the zipped folder to [CourseWorks](#). Also, bring a printout of your code to class.

Exercises:

You have been contacted by John Langford to help him solve the unsolved mysteries of the DaVinci Code.

1. **(20 points)** John and Sophie have found the key to open a safe deposit box at the Paris branch of the Depository Bank of Zurich, but they need to find the account's number. Luckily, they have a clue. The account number is a combination of the first 10 Fibonacci numbers (http://en.wikipedia.org/wiki/Fibonacci_number). Just to be sure, you want to generate the numbers of the famous sequence, to confirm that they got it right.
 - 1.1. Write a **for** loop to generate the first 1000 numbers of the Fibonacci sequence, and store them in a vector named *FibVec*.
 - 1.2. Do the same without using loops (HINT: looking online for closed form solutions of Fibonacci's series might help)
2. **(20 points)** Inside the box, John finds a criptex with 10 digits. In order to decypher it, he needs to plot some numbers, and asks for your help. Generate the following 10x10 matrix *M* (HINT: you may find the flipdim() function useful):

0	0	0	0	0	0	0	0	-1	2	
0	0	0	0	0	0	0	0	-1	2	-1
0	0	0	0	0	0	0	-1	2	-1	0
0	0	0	0	0	-1	2	-1	0	0	0
0	0	0	0	-1	2	-1	0	0	0	0
0	0	0	-1	2	-1	0	0	0	0	0
0	0	-1	2	-1	0	0	0	0	0	0
0	-1	2	-1	0	0	0	0	0	0	0
-1	2	-1	0	0	0	0	0	0	0	0
2	-1	0	0	0	0	0	0	0	0	0

3. **(20 points)** Inside the criptex there is a 3D plot. You must analyze it in order to find the coordinates of where exactly the Rose Line passes inside the Church of Saint-Sulpice, and find the next clue to solve the mystery.

(NOTE: each plot should be visualized in a separate figure)

- 3.1. Define $xvec$ and $yvec$ as vectors ranging from $-\pi$ to π (it's up to you how many elements you want them to have). Plot the surface $Z = X^2 \cdot Y^2$. Find the coordinates of the minimum value of the minimum value of the surface.
- 3.2. Compute and plot the contour function on Z, visualizing also the levels labels.
- 3.3. Re-plot Z, together with all the planes parallel to the X-Y which were used by the contour function (i.e. plot one plane per level from point 3.2, the same way as in slide 41 of Lecture 2).

4. **(20 points)** Silas got to the church before you, but luckily that clue was just a deception. The real information about who to contact to solve the mystery is contained in another graph, which was given to John by his friend Sir Leigh Teabing.

Load the file *surface.mat* into MATLAB

- 4.1. Plot the points of the surface *im* in the x range [1 116] and y range [1 156]
- 4.2. Apply the colormap to be 'Bone' and replot the surface
- 4.3. Define and apply your own colormap
- 4.4. Who does the surface represent?
- 4.5. What part of the face has lowest values among mouth, eyes, cheeks and forefront?

Plot the surfaces with the 3 colorbars in the same figure, arranged as a 3x1 column. What happens to the colorbars when you do that? Look online for a solution to the problem.

5. (20 points) Wow! You would have never imagined that *that* person was involved! John somehow manages to get in contact with that person, who gives him the final clue to solve the mystery. It's a race against time between you and Silas. Whomever computes the function $y(x) = \pi e^{-x}$ in 10001 equally spaced points in the interval $x \in [0,100]$ in the most efficient way, will unveil the secret! Below you can see Silas' code.

5.1. Modify it in the most efficient way you can

5.2. Compute the difference in execution time between Silas' and your implementation.

```
x=0;
for k=1:10001;
    y(k) = exp(-x)*pi;
    x = x + 0.01;
end
```