**MATLAB Programming: Midterm Two**

Student\_Name: 蔡育呈

Student\_ID: 0716074

Date: 2020/5/21

Problem 1 (2pt): \_\_88\_\_ (WORD Count. Must fill it.)

Problem 2 (2pt): \_\_84\_\_ (WORD Count. Must fill it.)

Problem 3 (3pt): \_\_112\_ (WORD Count. Must fill it.)

Problem 4 (3pt): \_\_120\_ (WORD Count. Must fill it.)

===========================================================================

**Problem 1:**

There are two curves which are plotted in the polar and Cartesian coordinate systems, respectively.

Write a program to reproduce the animation result which **must be similar** to the example. The colors can be different from the example.

Curve 1 (must be in the polar coordinate system):

a = 10;

x = [-4\*pi:0.1:-eps eps:0.1:pi\*4];

r = a \* sin(x) / x;

Curve 2 (must be in the Cartesian coordinate system):

a = 5; b = 15; t = [0:0.1:6\*pi];

x = (a+b)cos(t) – b\*cos((a/b+1)t)

y = (a+b)sin(t) – b\*sin((a/b+1)t)

**Report:**

First, I used subplot to divide the figure into two parts. In the first part I used polar, then turn it into Cartesian coordinates and fill it. The second part, I first added 4\*pi to make the domain of theta match the domain of t, then plot and fill. I used for loop and clf to update my figure, which gains new parts of the curves at each loop as well as keep the previous ones. Finally, I set the axis and used pause to show the animation.

===========================================================================

**Problem 2:**

Set t = -10:0.025:10. The step size can be adjusted but the curves must be smooth.

We can interpolate the two functions as follows:

x(t) = (1-k) x1(t) + k x2(t)

y(t) = (1-k) y1(t) + k y2(t).

k can be computed in two ways. Thus, we produce the animation process to interpolate the two functions in two plots. Let h = 0:0.01:1. The two ways are:

k = 2\*h\*sin(h\*2\*pi), and k = sin2(k1\*pi/2), where k1 = 2\*h\*sin(h\*2\*pi).

Ask to press ENTER before the animation is generated. The animation result must be similar to the example. There must be two plots (use subplots). Each plot shows one interpolation result. The animation must be smooth and interactive. Make sure that the figure window is on top.

**Report:**

The problem asked us to draw two parametric functions. First, I typed the functions and waited for the user to press ‘Enter’ key to trigger the animation. The animation is finished by pause and for loop, and similarly using clf to update the figure. Then I used subplot to divide the figure. I first drew the original function which would be static on the figure. Finally, in each plot, I combined two functions linearly to interpolate two curves with different k, and set axis.

===========================================================================

**Problem 3:**

Initial condition: t = 0, v = [-1 7] and p = [50 3]. p is a function of time.

Dt = 0.05.M = 10000, m= 1. d = 0.01. The frame number, *i*, is increased by one after all the rules are applied once. **The parameters can be modified**.

The termination condition is t >= tmax, where tmax = 100.

Collect the most recent N sample points of the planet trajectory. Compute the length of the curve segment formed by these N points. N must be at least 8.

Plot the length of the curve segment vs the frame number, *i*.

If the length cannot be computed for a small i, the plot is skipped.

**Report:**

The problem is separated into two parts. First, I asked the user to input the number of points. After setting the initial situation of the dynamic system, I used while loop to calculate the force, acceleration, velocity and position of every single moment. Similarly, the animation is fulfilled by pause, for loop and clf, and I also used subplot to divide the figure into two. As for the first figure, I plotted the track and then plotted the n recent points. For the second, I calculated the sum of the curve length of the n points and plotted the statistic chart, with the x-coordinate number of points, and sum be the y-coordinate.

===========================================================================

**Problem 4:**

This program manipulates the image, tmp.png. You can resize the image as long as the objects are clear in the image. You press keys to manipulate the image. Note that if the image is brightened up, its effect should be on when the image is being shifted.

You should use the template: m02\_4.m. Change file name.

The template only uses the key ‘j’ only.

The results must be similar to the results in the demo video.

Show the options for the key usage as follows:

j) Shift the image from right to left by 10 pixels per step

k) Shift the image from left to right by 10 pixels per step

n) Shift the image from right to left by 40 pixels per step

m) Shift the image from left to right by 40 pixels per step

b) Turn on or off a spot light on the center. Raise or reduce the intensities of pixels in a circular ring (formed by two circles). The region inside the smaller circle is much brighter than the region between the two circles.

q) Show student name and ID. Then quit the program.

**Report:**

The problem needs to fulfill five actions about shifting images and turning on/off the spotlight. As for the part of shifting, for instance, I’d like to shift right. Then I place the 1:i column to the terminal of the new image, and place the i+1:end column to the start. And subtract one from i once a column is shifted. Other shifting processes are similar, only changing a few parameters, including the number of pixels changed each time, and the shifting direction. And for the last, I used meshgrid to produce a coordinate system, with (0,0) the center. Then calculate the distance of each point to (0,0), and distributing them into three ranges, which would show different brightness in the image.