Amy Yu
UID: 905138432
CEE/MAE M20
October 29, 2019

**Homework 4**

## 1. Correlation Coefficient of Populations of Lynx and Hare in One Period

### 1.1 Introduction

The goal in this problem is to use arrays and the given Lotka-Volterra equations to find the population of hare and lynx over 40 years. The populations over time are plotted and printed as a figure. Additionally, one period's worth of data is extracted from each species' population and the Pearson's linear correlation coefficient is calculated between the two species.

### 1.2 Model and Methods

The discretized equations using the forward Euler method are used to calculate the population of hare and lynx, and they were calculated as follows:

$$\frac{x_{k+1} - x_k}{\Delta t} = 0.4x_k - 0.018x_k y_k$$
$$\frac{y_{k+1} - y_k}{\Delta t} = -0.8y_k + 0.023x_k y_k$$
$$x_{k+1} = x_k + \Delta t(0.4x_k - 0.018x_k y_k)$$
$$y_{k+1} = y_k + \Delta t(0.4x_k - 0.018x_k y_k)$$

The values of the given coefficients are set as variables in order to prevent them from accidentally being changed. The same was done with the initial and final times and delta T. Three one dimensional arrays of zeros and size Nsteps were created to represent the population of hare, the population of lynx, and the time. The number of total steps was calculated and used in a `for` loop. The loop allows us fill the arrays and also calculate the period.

The period was calculated in the `for` loop by creating a variable to store the time that the maximum population occurs for one of the species. From the graph, we can estimate that the first peak occurs before the time is at 10 years, so this is set as a condition in the `if` statement. The max value is updated as the `for` loop runs. The same thing is done for the second peak, this time estimating that the second peak occurs before 25 years. Then, the difference in time between the first and second peaks is the period.

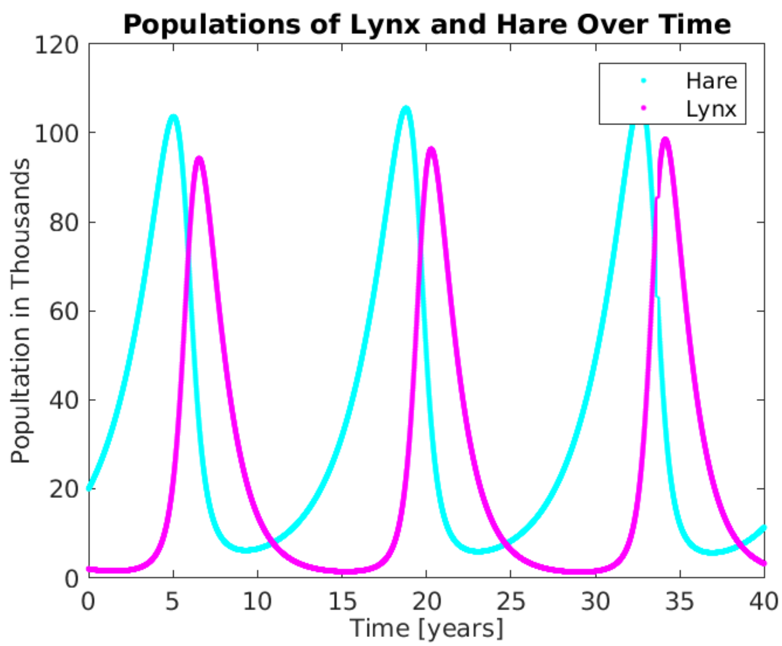
The curve for each species is plotted on the same graph, set to a different color, and a legend is created. The title and axes labels are also created. The `save as` function is used to save the figure as an image, and the value of Pearson's linear correlation coefficient is printed to the screen.

### 1.3 Calculations and Results

When the program is executed, the following is printed to the screen:

```
Pearsons linear correlation coefficient is -0.003051
```

The following figure is printed to the screen:




### 1.4 Discussion

Based on Pearson's linear correlation coefficient, lynx and hare do not have a high linear correlation, since the value is very close to 0. The figure shows that the populations of lynx and hare oscillate periodically, and one lags slightly behind the other.

## 2. Pendulum Simulation

### 2.1 Introduction

The goal in this problem is to simulate a pendulum that is displaced a certain angle from its resting position using the given kinematic equations. A damping force is also taken into account, and both simulations are written using the explicit Euler method and semi-implicit Euler method.

### 2.2 Models and Methods

The discretized equations from the kinematic equations of a pendulum are calculated as follows:

**Explicit**

$$\frac{\omega_{k+1} - \omega_k}{\Delta t} = -\frac{\mathrm{g}}{\mathrm{L}}\sin\left(\theta(\mathrm{k})\right)$$

$$\frac{\theta_{k+1} - \theta_k}{\Delta t} = \omega(k)$$

$$\omega_{k+1} = \omega_k - \Delta t\left(\frac{\mathrm{g}}{\mathrm{L}}\sin\left(\theta(\mathrm{k})\right)\right)$$

$$\theta_{k+1} = \theta_k + \Delta t(\omega(k))$$

**Semi-Implicit**

$$\frac{\omega_{k+1} - \omega_k}{\Delta t} = -\frac{\mathrm{g}}{\mathrm{L}}\sin\left(\theta(\mathrm{k})\right)$$

$$\frac{\theta_{k+1} - \theta_k}{\Delta t} = \omega(k+1)$$

$$\omega_{k+1} = \omega_k - \Delta t\left(\frac{\mathrm{g}}{\mathrm{L}}\sin\left(\theta(\mathrm{k})\right)\right)$$

$$\theta_{k+1} = \theta_k + \Delta t(\omega(k+1))$$

The script first sets the given variables and times as variables, and calculates the number of steps. Then, six arrays of zeros and size Nsteps are created, with three for the explicit Euler method and three for the semi-implicit Euler method. The three arrays represent the values of omega, theta, and the time. The equations above are used in a `for` loop to compute each method, and both methods are plotted on the same graph.

For the second part of the problem, a damping force is taken into account. The discretized equations are calculated as follows:

**Explicit Damping**

$$\frac{\omega_{k+1} - \omega_k}{\Delta t} = -\frac{\mathrm{g}}{\mathrm{L}}\sin\big(\theta(\mathrm{k})\big) - \omega(\mathrm{k})\mathrm{L} \times \mathrm{d}$$

$$\frac{\theta_{k+1} - \theta_k}{\Delta t} = \omega(k)$$

$$\omega_{k+1} = \omega_k + \Delta t\left(-\frac{\mathrm{g}}{\mathrm{L}}\sin\big(\theta(\mathrm{k})\big) - \omega(\mathrm{k})\mathrm{L} \times \mathrm{d}\right)$$

$$\theta_{k+1} = \theta_k + \Delta t(\omega(k))$$

**Semi-Implicit Damping**

$$\frac{\omega_{k+1} - \omega_k}{\Delta t} = -\frac{\mathrm{g}}{\mathrm{L}}\sin\left(\theta(\mathrm{k})\right) - \omega(\mathrm{k}+1)\mathrm{L} \times \mathrm{d}$$

$$\frac{\theta_{k+1} - \theta_k}{\Delta t} = \omega(k+1)$$

$$\omega_{k+1} = \omega_k + \Delta t\left(-\frac{\mathrm{g}}{\mathrm{L}}\sin\left(\theta(\mathrm{k})\right) - \omega(\mathrm{k}+1)\mathrm{L} \times \mathrm{d}\right)$$

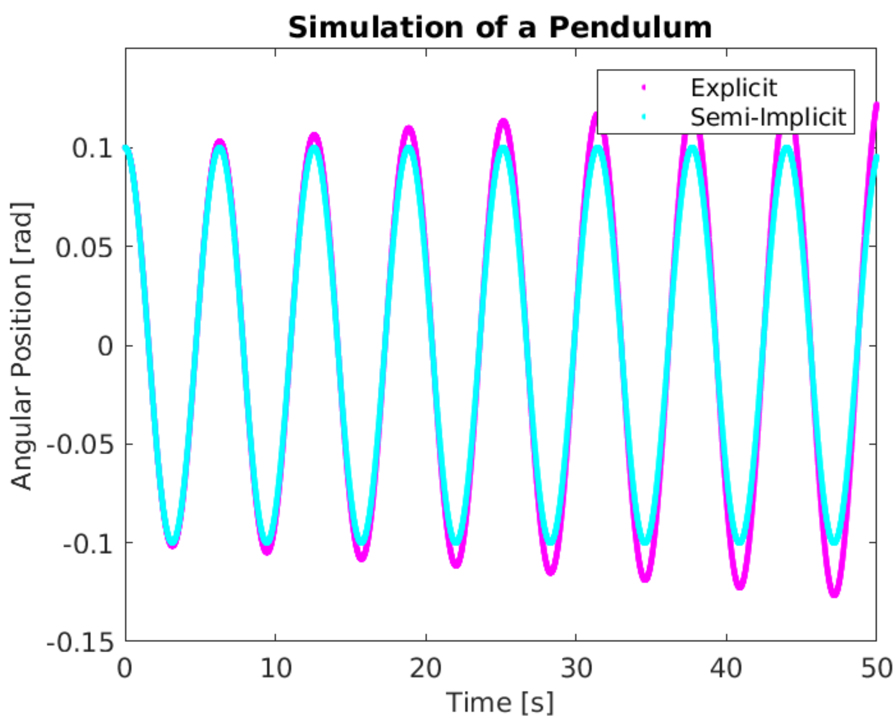
$$\theta_{k+1} = \theta_k + \Delta t(\omega(k+1))$$

The script is the same as the one in the first part of the problem, but the discretized equations are different.

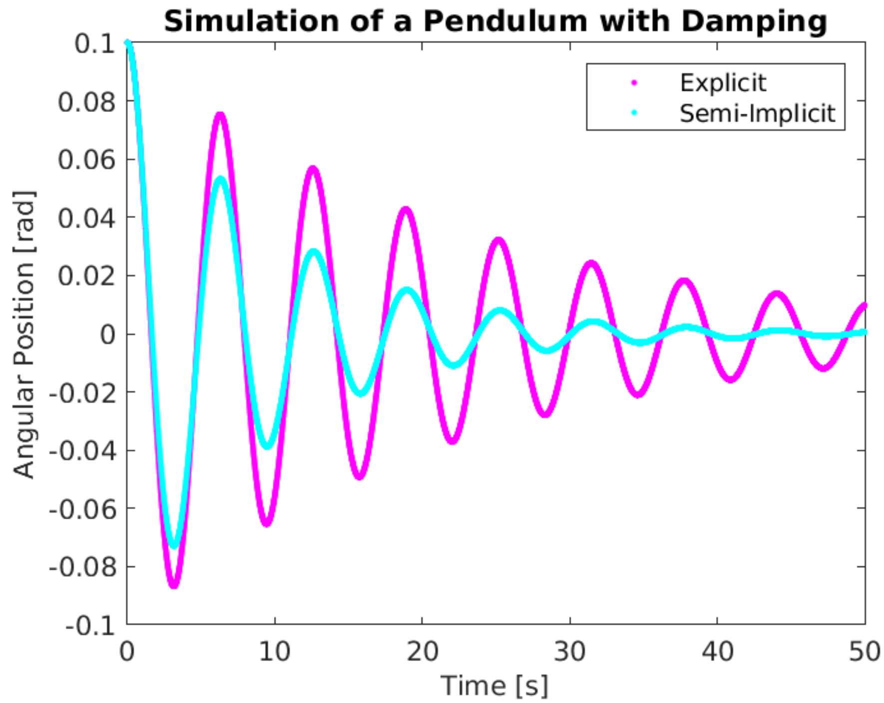
Lastly, using semi-implicit Euler only, plots was created from t = 0 to t = 100s for three different values of d, which affects the damping. The code for the semi-implicit method was taken from the previous part, and the value of d was varied to produced three plots.

**2.3 Calculations and Results**

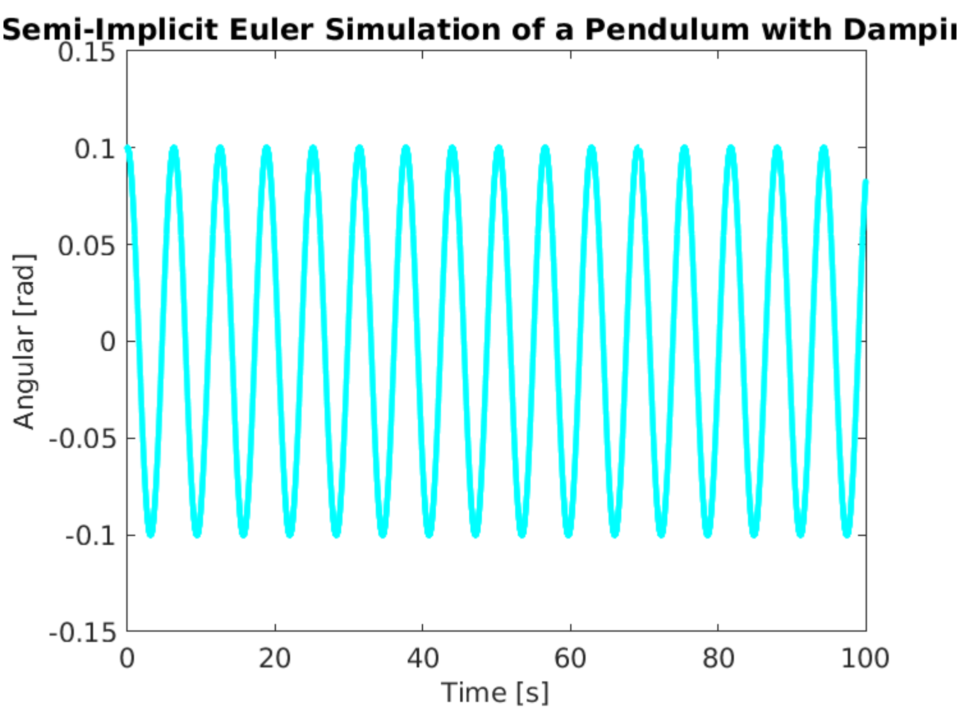
In the simulation without damping, the following figure is printed to the screen:



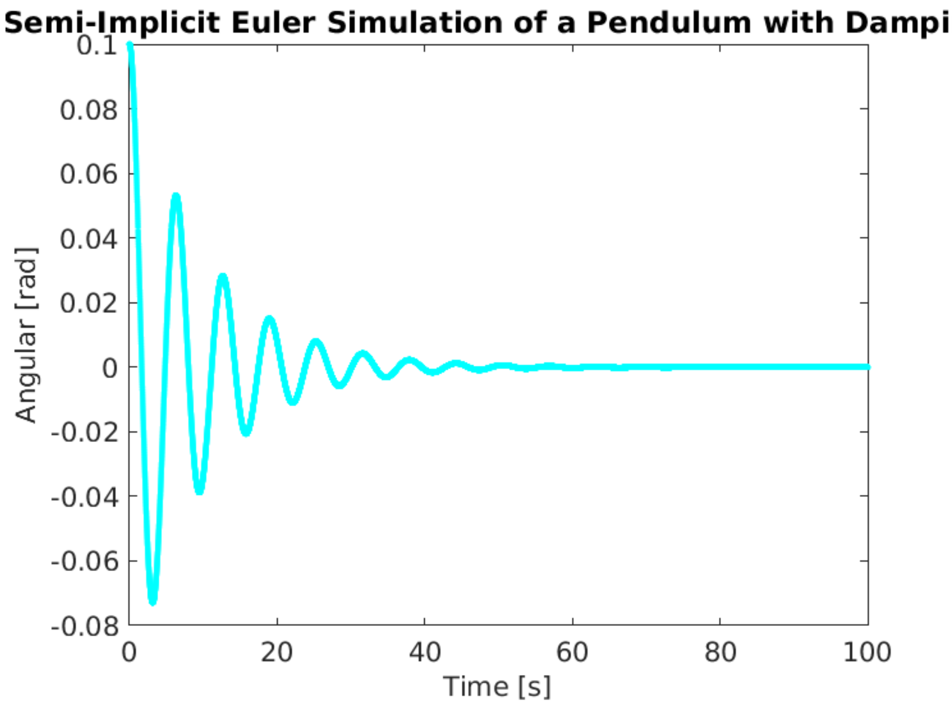

In the simulation with damping, the following figure is printed to the screen:



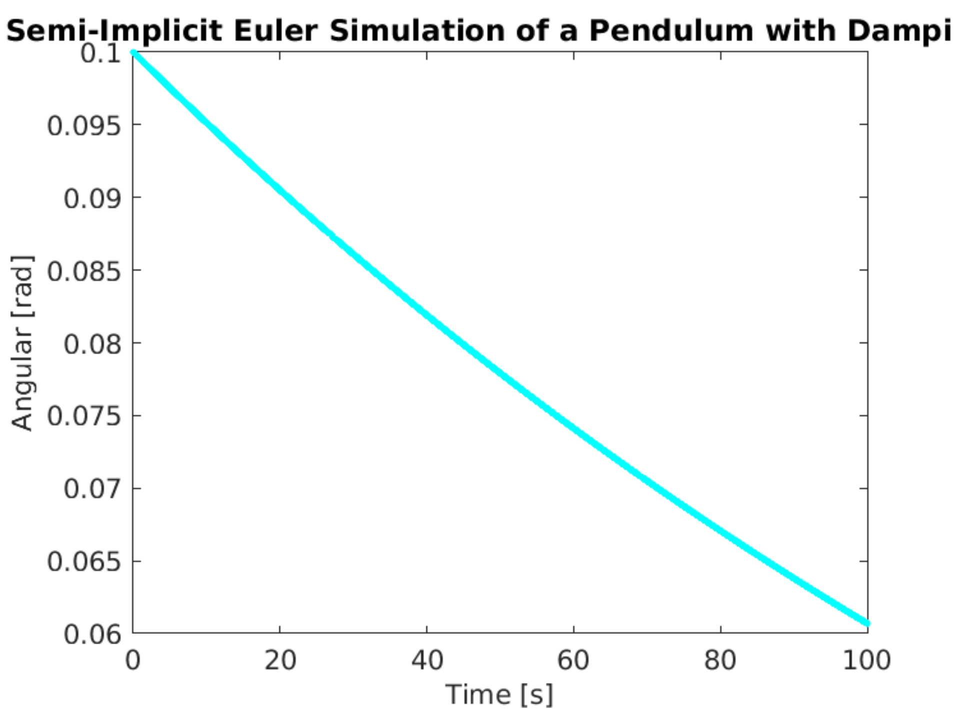

When d is set to 0 using the semi-implicit method, the following figure is printed:



When d is 0.01, the following figure is printed:




When d is 10, the following figure is printed:



## 2.4 Discussion

In the figure without damping, it can be seen from the figure that energy is added to the system in the explicit method, as the amplitude of the oscillations is increasing over time. The semi-implicit method provides a more accurate simulation, as it stays at the same amplitude.

For the figure with damping, d is set to 0.01. This should simulate underdamping, which should look more like the semi-implicit curve. Additional energy is added in the explicit method, so the amplitude is larger than that of the semi-implicit figure.

Using the semi-implicit method and varying the values of d, we can see than d=0 is undamped, d=0.01 is underdamped because it is less than 1, and d=10 is overdamped because it is greater than 1. In the undamped simulation, the amplitude of the oscillations stays the same throughout the 100 seconds. In the underdamped simulation, the oscillations get smaller and die out by 60 seconds. In the overdamped simulation, there is no oscillation at all and it slowly returns to its equilibrium position.