Ben Zeman

PHY 6860

6 March 2019

Midterm Problem 1

This problem was very similar to HW #3, with the goal being to analyze an oscillating pendulum analytically, numerically, and with linear and nonlinear ODES. For this problem, instead of the Euler-Cromer method, I used the Runge-Kutta 4th order method to calculate the angle and angular velocity of the pendulum over time. I varied the driving acceleration and driving frequency. I compared the amplitudes and phase angles to the pendulum's angular frequency, and I found Lyapunov exponents.

To test the accuracy of the program, I needed analytical solutions for resonance frequency, phase angle, and amplitude to compare the results to. I had several initial conditions:

Gravity (g) = 9.8   length (l) = 9.8   gamma = .25 driving acceleration (alpha) = .2 change of time (dt) = .01

I could use these to find other useful constants:

phase frequency = sqrt(g/l)
driving frequency (at resonance) = sqrt(phase frequency^2 - (2*gamma^2))

Once I had these constants, I could use these equations to find the constant amplitude and phase angle of the oscillation:

amplitude = driving acceleration/sqrt(((2*gamma*driving frequency)^2) + ((phase frequency^2-driving_frequency^2^2))
phase = -arctan((2*gamma*driving frequency)/(phase frequency^2 - driving frequency^2))

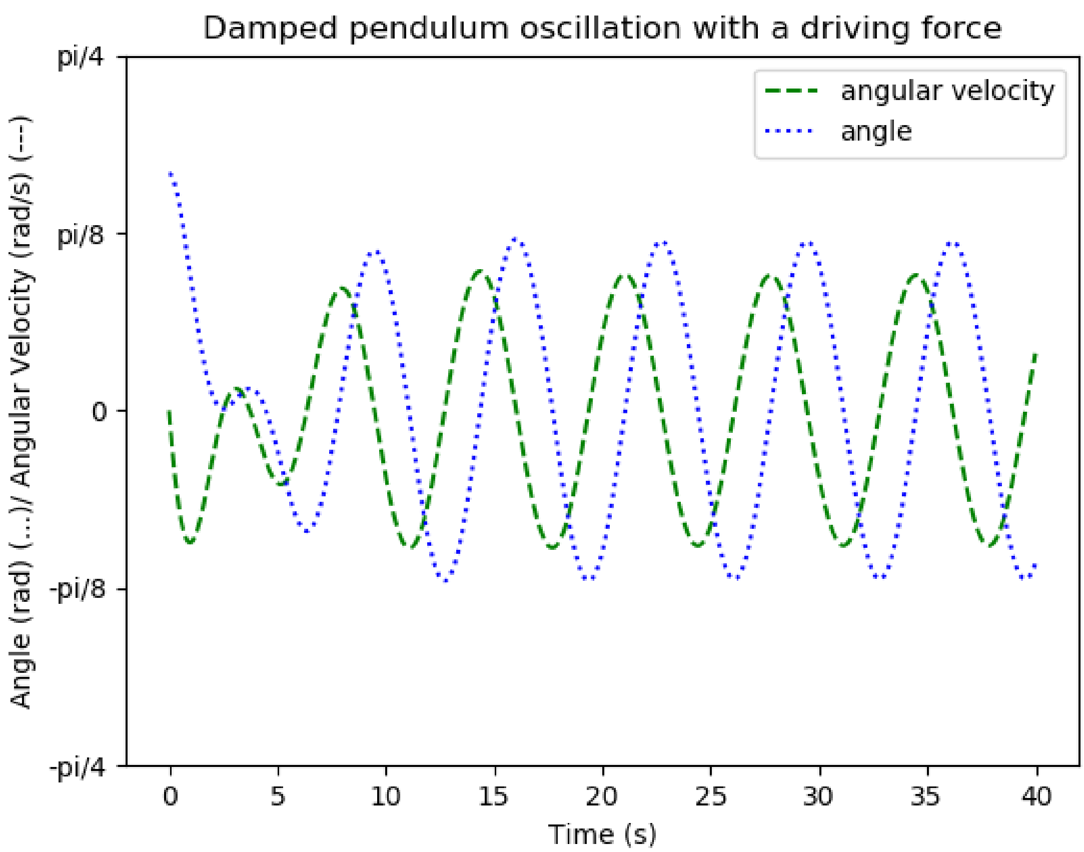
From the output of my program:

```
The analytical amplitude is 0.13pi
The analytical phase angle is -0.42pi radians
```

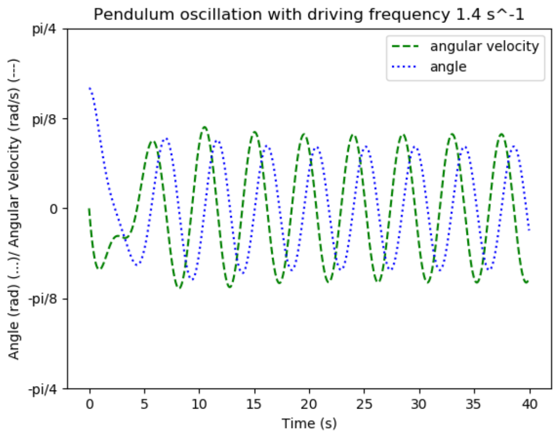
After obtaining these numbers, I made a numerical model of the same oscillation with the same initial conditions. I compared my driving frequency, phase, and amplitude to my analytical solution. Then, I made more numerical models to investigate the relationships described in my intro. Here is my process:

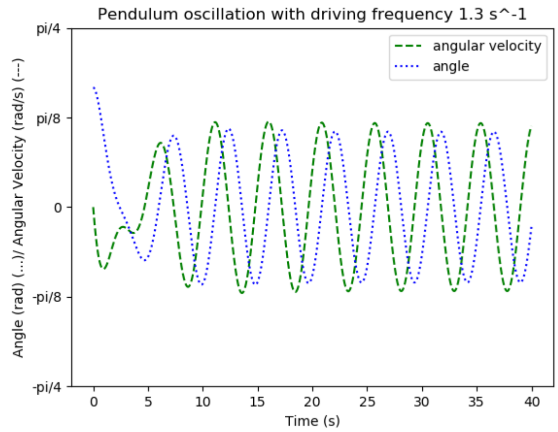
- Used my calculated driving frequency
- Initialized the angle and angular velocity
- Made a while loop to append the two variables for each time-step
    - Each while loop ran long enough (40 seconds) for the oscillations to reach their steady states
    - Updated variables with the 4th order runge-kutta method
        - Equations included k variables, which were based on functions of angle, angular velocity, and time
    - Used the small-angle approximation, substituted the sine of the angle for the angle itself
    - Made an if/else statement to keep the angle between -pi and pi

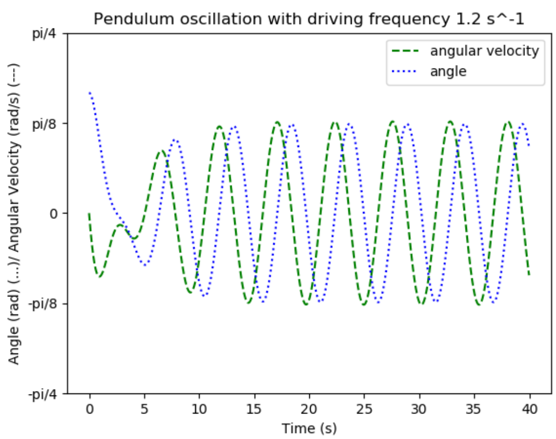
- Plotted the angle and angular velocity in blue and green with different line styles

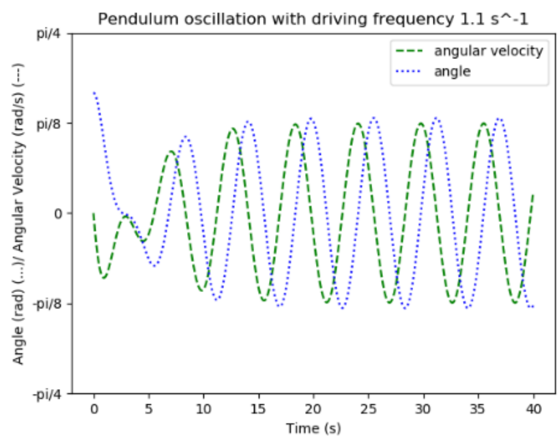


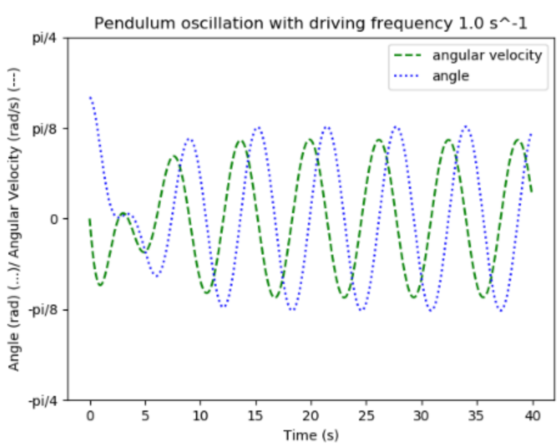
- Amplitude can be extracted at local angle minimum in the steady state
    - Amplitude at 33 sec is about pi/8 rad, very close to analytical solution of 0.13pi
- Phase can be extracted from the same local minimum, using equation:
    - Phase=2pi(integer)+1.5pi-(frequency)t
    - About (2(integer)+1.5)pi-(.935)(33)
    - Guessed the integer to make phase within -pi/2 and pi/2 to be 4
    - Phase is about 8pi +1.5pi –(.935*33) = -0.326pi
    - My analytical calculation was -0.42pi, which is relatively close
- The small angle approximation accurately models the amplitude of this oscillation, but there is significant error when modeling the phase angle
- Next, I made a for loop to model similar oscillations with 10 different driving frequencies
    - My analytical frequency was not used
    - Used same runge kutta method
    - Graphed 10 separate plots for each frequency

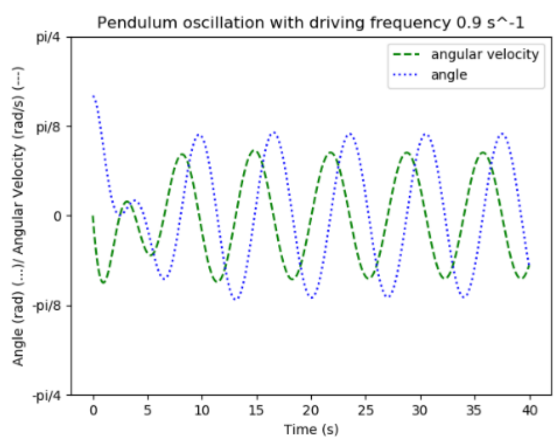

Pendulum oscillation with driving frequency 1.4 s^-1
angular velocity
angle
Angle (rad) (...)/ Angular Velocity (rad/s) (---)
Time (s)

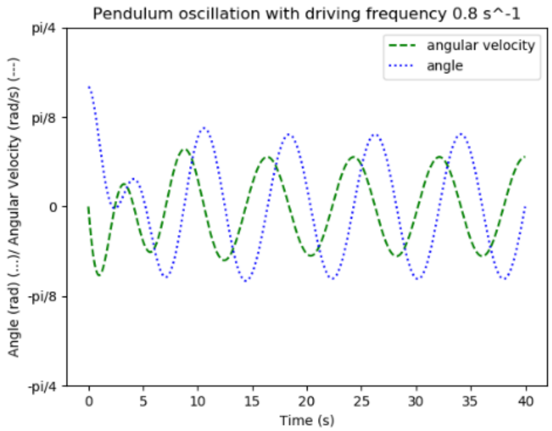

Pendulum oscillation with driving frequency 1.3 s^-1
angular velocity
angle
Angle (rad) (...)/ Angular Velocity (rad/s) (---)
Time (s)

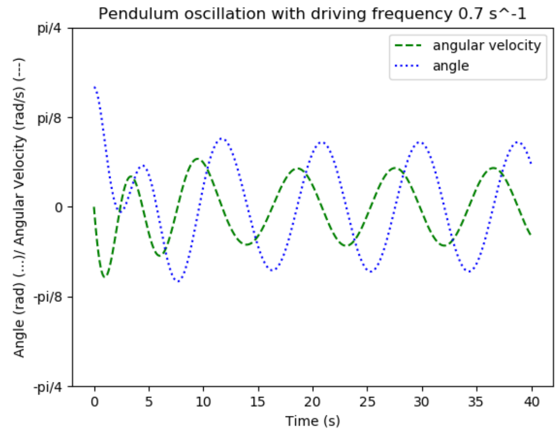

Pendulum oscillation with driving frequency 1.2 s^-1
angular velocity
angle
Angle (rad) (...)/ Angular Velocity (rad/s) (---)
Time (s)

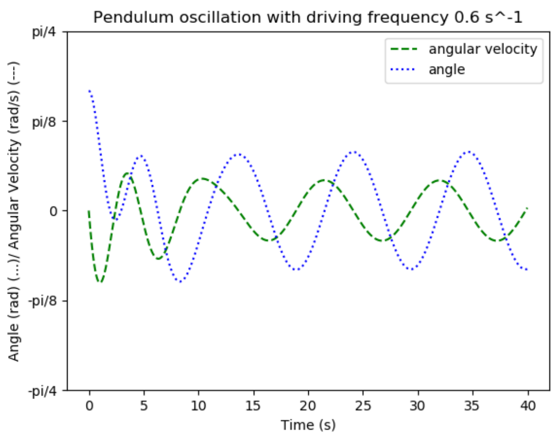

Pendulum oscillation with driving frequency 1.1 s^-1
angular velocity
angle
Angle (rad) (...)/ Angular Velocity (rad/s) (---)
Time (s)

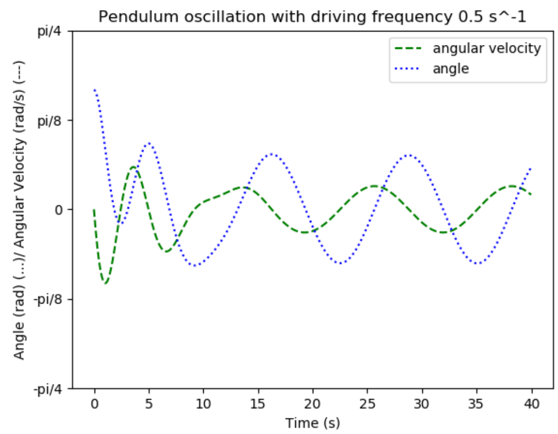

Pendulum oscillation with driving frequency 1.0 s^-1
angular velocity
angle
Angle (rad) (...)/ Angular Velocity (rad/s) (---)
Time (s)


Pendulum oscillation with driving frequency 0.9 s^-1
angular velocity
angle
Angle (rad) (...)/ Angular Velocity (rad/s) (---)
Time (s)

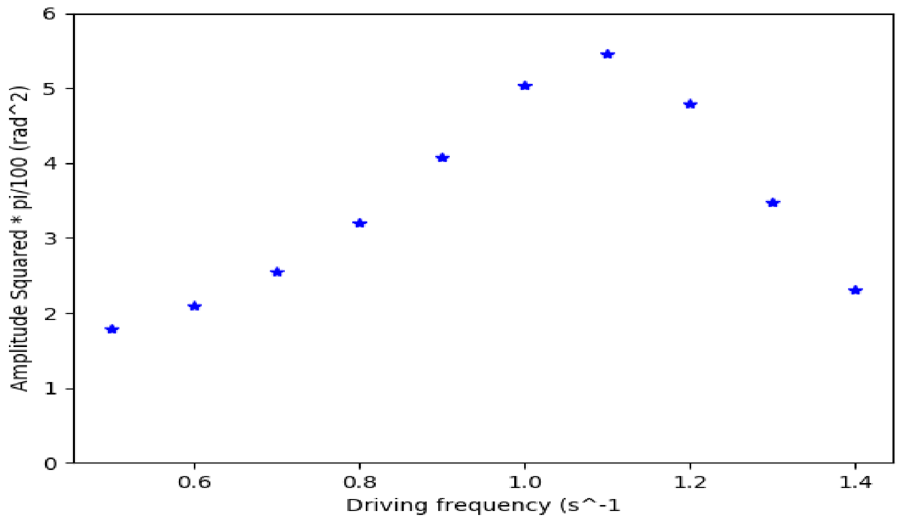




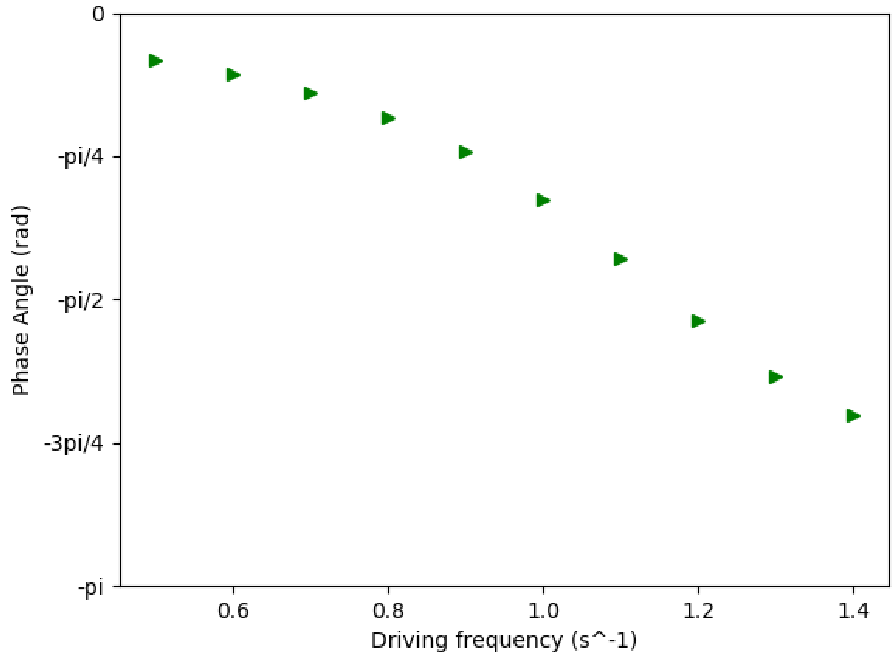




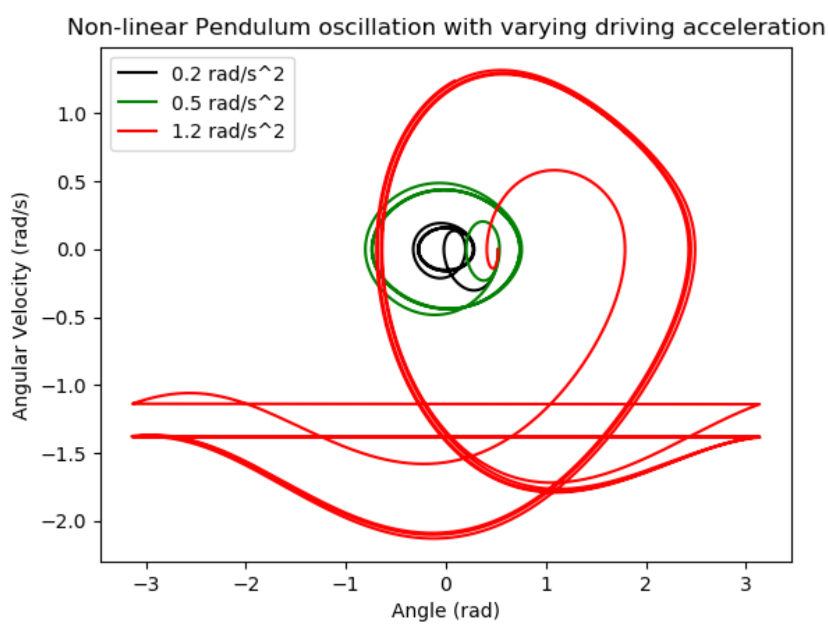
- As the graphs show, the higher frequencies reach the steady state quicker due to the driving force
  - It is easier to extract important values from higher driving frequencies, as the portions of the graph before the steady states isn't very useful
- Within the for loop, I used the local minimums at t>28 seconds to record the amplitudes for each of the oscillations
  - I plotted the amplitudes squared vs their corresponding driving frequencies



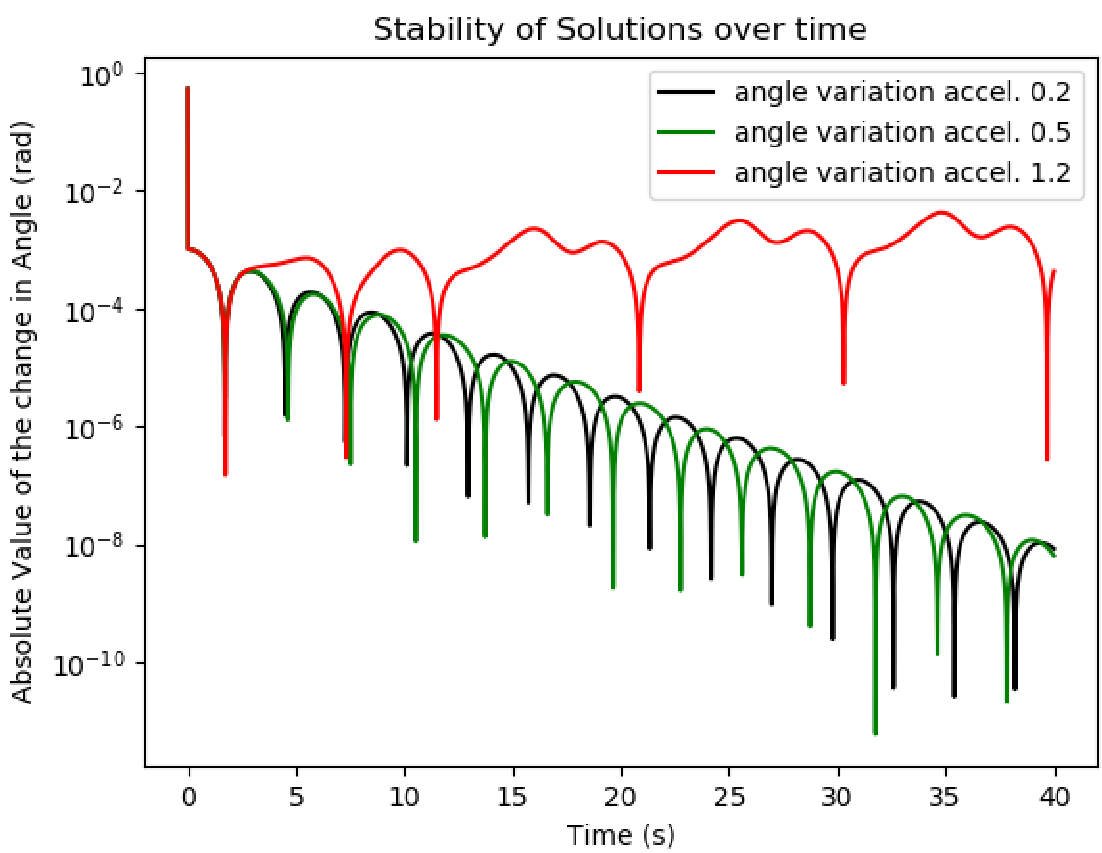
- From this graph, I could estimate the gamma value for these oscillations
  - I estimated the width of the plot at half the maximum
    - Half max is close to 2.75 rad^2
    - At this amplitude squared, the frequencies are about 0.75 and 1.35
    - The gamma value is estimated at 0.6, which is quite far from our given value of 0.25
    - The reason for this is likely because the low-frequency oscillations were close to steady state, but it was hard to tell if they had completely reached steady-state, so the amplitudes on the left side of the graph are somewhat inaccurate
    - The problem is, if we ran the program long enough to get all oscillations to a perfect steady state, maybe for 100 sec, it would be hard to determine phase angles and times for the amplitudes of the high-frequency oscillations
- Then, I graphed the phase angle for each frequency



- This plot shows a decrease of the phase angle, with absolute increase
  - The analytical plot for this relationship has asymptotes at 0 and -pi, which appear to exist here too, showing that this plot is accurate
- Made a similar for loop, but varying acceleration rather than frequency
  - Driving frequency constant at 2/3
  - Runge kutta method, but nonlinear
    - In angular frequency equation theta is replaced by sin(theta)
  - Angle vs velocity plotted for each alpha value on same graph



- Graph shows a more controlled relationship for the 0.2 and 0.5 accel. plots, repetitive circling
    - The 1.2 accel plot appears uncontrolled, expanding infinitely, which shows that chaos is likely
- Finally, I wanted to show evidence of chaos, so I repeated the same for loop as the last one, but created theta 2 and omega 2, which are exact duplicates of theta and omega but with initial angles 0.001 rad apart.
    - The while loop appended theta, omega, theta 2, and omega 2, with runge kutta
    - I made an array of the absolute value of the change in angle, omega 2 – omega, for each iteration
    - I plotted this change over time



- This plot appears to show linear trends on the log scale for accelerations 0.2 and 0.5, but shows chaos for 1.2
    - The Lyapunov exponents can be estimated by the variation of the angle change over time
    - The 0.2 and 0.5 plots both show a change of log(1E-4) rad at 8 seconds and a change of log(1E-6) rad at 25 seconds.
        - This allows me to estimate a slope of (log1E-6 – log1E-4)/17
        - The Lyapunov exponent is estimated at -2/17
        - The negative exponent shows no chaos

    - The 1.2 plot is harder to analyze but shows an approximate change of log(1E-3) rad at 15 seconds and a change of log(1E-2) rad at 35 seconds.
        - The Lyapunov exponent is estimated at 1/20
        - The positive exponent shows chaos

This program was very similar to HW #3, but gave me a chance to try a new integration method, the runge kutta method. Also, when I calculated amplitude and phase in my program, I was able to check how accurate my program was in a more acceptable way than by looking at a graph. Also, the gamma estimation allowed me to realize that even though the oscillations appeared to be in steady state, they really weren't yet.