Jessica Hamilton
Computational
Exam 02
11-4-2019

1. Types of Equations
Write an example of each type of equation listed below. Your examples do not need to be from real systems.

(a) Linear equation
$y(x) = 3x + 9$

(b) Non-linear equation
$y(x) = cos(x) + 2 * sin(x)$

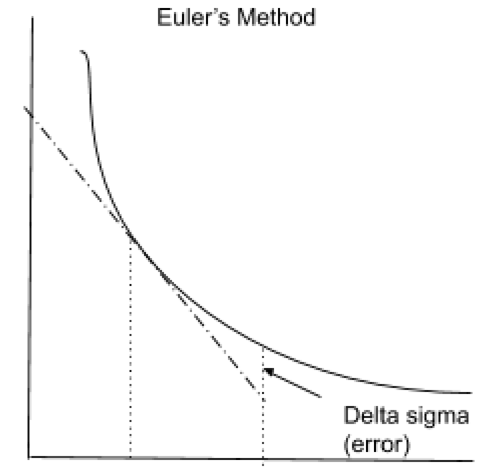
(c) 1st-order, linear differential equation
$\frac{dy}{dx} + 4y = 3x + 4$

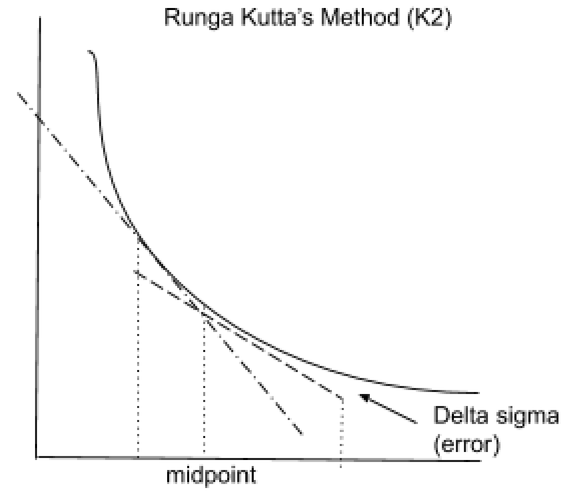
(d) 4th-order, non-linear differential equation
$\frac{d^4y}{dx^4} - y^2 + 3y = 12$

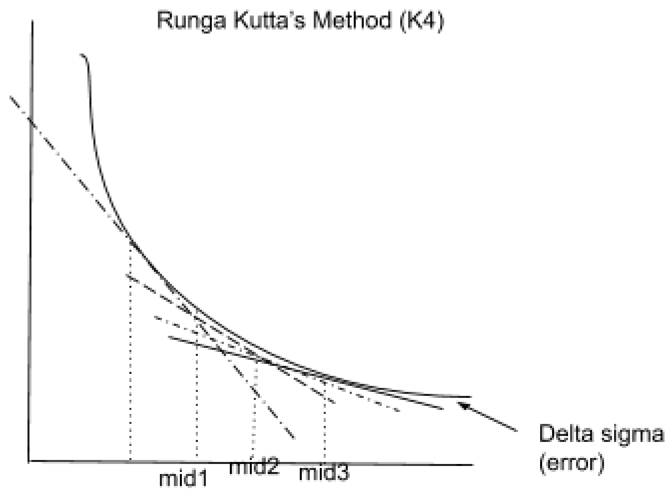
(e) Two coupled, first-order linear differential equations
$\frac{dx}{dt} = 2x + 3y \ and \ \frac{dx}{dt} = 4x + 12y$

2. ODE Algorithms
Explain why (and how) a 2nd-order Runge-Kutta (RK2) method is more precise (i.e., it has a smaller numerical error) than Euler's method for solving a 2nd-order non-linear differential equation, assuming equal step sizes. Use figures to aid your explanation, if necessary. What is different between an RK2 and an RK4 method and why is RK4 more precise?

The Runge-Kutta (RK2) method is more precise than the Euler method for solving 2nd-order non-linear differential equation. When using the RK2 method, you are using the slopes defined at the midpoint which is determined by the Euler method and the previously determined slope. This provides a weighted slope. From the weighted slope the next position and velocity values are determined. Since the RK2 method utilizes the midpoint, the delta sigma (error) will be less than that of the Euler method. The RK4 method determines the midpoint from the previous two slopes (k2 and k3), midpoint(k4) of the midpoint(k3) of the midpoint(k2). This will take the slope for the next step, much closer to the original function, minimizing the error even more. The Euler method makes use of the slope as well, but at the given (previous) point only. It then estimates the next position and velocity values based on that slope. See graphs below for visual representation of each method, Euler's, RK2, and RK4 respectively.







3. Physical Pendulum

Imagine a mass, m, suspended from a massless rod of length, l. You raise the mass so that the rod makes an angle θ with the vertical (i.e., with the mass's resting position). Write a code to solve for the mass's motion over time once it is released.

(a) Setup the problem by first sketching the system and writing down the relevant equations.

Please see notes attached for drawing and relevant equations.

(b) Select an algorithm (or algorithms) to solve the problem. Explain and justify your choice of algorithm.

The algorithm I chose to use is the Euler's Method. The error is slightly larger than other potential methods, but this is a more straightforward method for the purpose of this problem, it is okay. This algorithm will allow us to find the position over time when not given any other information to begin. We can use relevant equations to determine position in polar and cartesian coordinates, and velocity.

(c) Write pseudocode outlining how you plan to solve for the mass's motion.

So to begin:

Import packages needed for problems (numpy and matplotlib)

Then define variables needed:

G , l, omega, h, initial theta, initial d_theta

Then initialize the arrays needed:

Time, theta, d_theta, x,y, v_x, v_y,, other ones that may be needed

Now begin for loop through time.

Find new d_theta

Find new theta

Append both to apprays

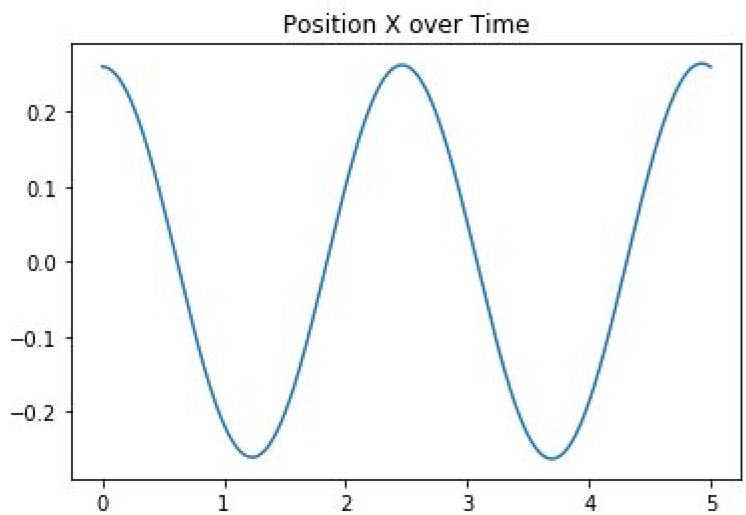
Convert to cartesian for both position and velocity

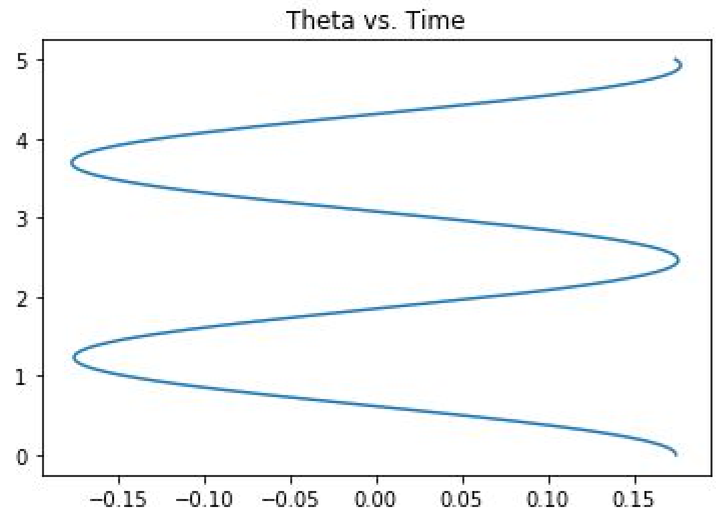
Calculate period of oscillation

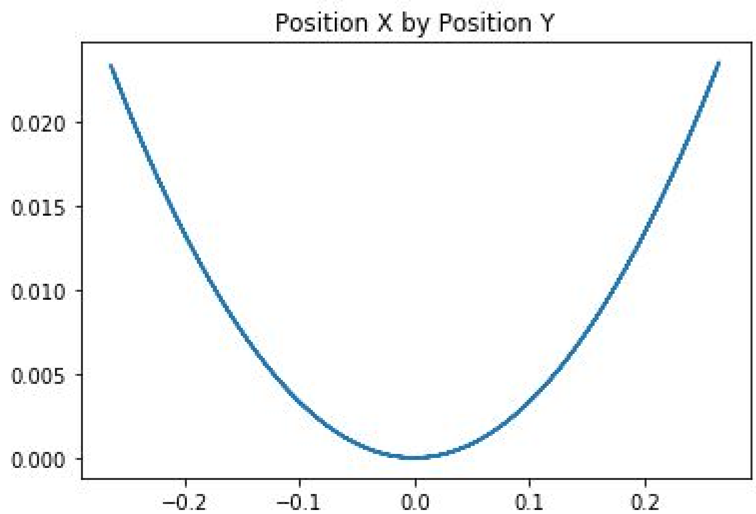
Then plot everything

(d) Make a plot of the mass's position in Cartesian space and also plot the y-velocity against the x-velocity. Describe your figures and explain why they make sense.

Below you can see the position of the pendulum over time:

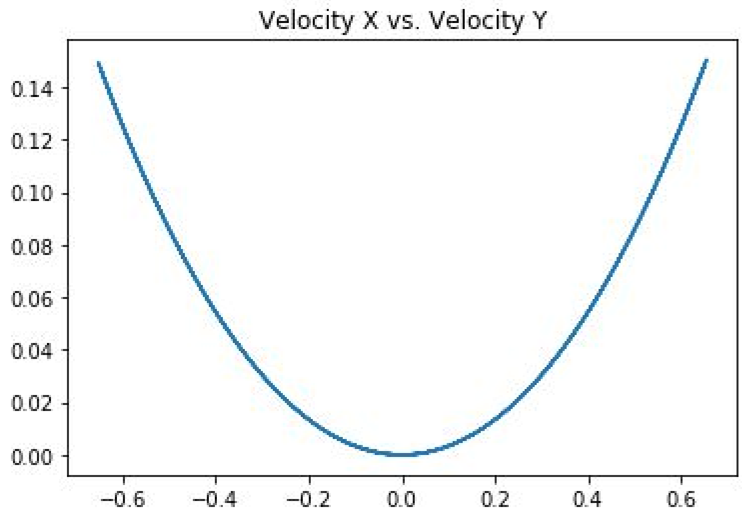






This graph is showing the position of the pendulum over time. As time progresses, you can see the changes in x are oscillatory (up and down), which is why the shape repeats. You can see that there are oscillations in both graphs. This graph is what we would expect to physically see when the pendulum oscillates back and forth in cartesian space. As time progresses, the position in x will slowly vary from positive to negative and back again. You can see this is also the case in the position x versus position y graph which the two are directly proportional. I have also included the Theta versus time, which is what coordinates I initially work in.

Below is the X Velocity vs Y velocity:



The graphs showing the velocity in x versus the velocity in y shows how the velocity in each direction is proportional to one another. As the velocity in x increases, the velocity in the y also increases.

Initial period for this example is: Period: 2.458173089805204

(e) Find the period of your pendulum for a few different initial angles and rod lengths. How do your results compare with the theoretical expectation for a simple pendulum: T = 2π s ` g When do your results begin to differ significantly from the theoretical expectation?

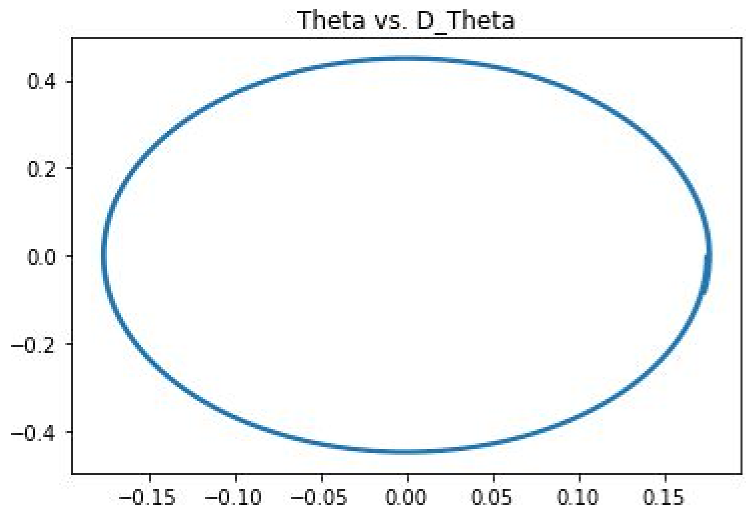
Results of the period for a few different angles and rod lengths:

Periods for changing Thetas: 2.458173089805204 2.458173089805204 2.458173089805204

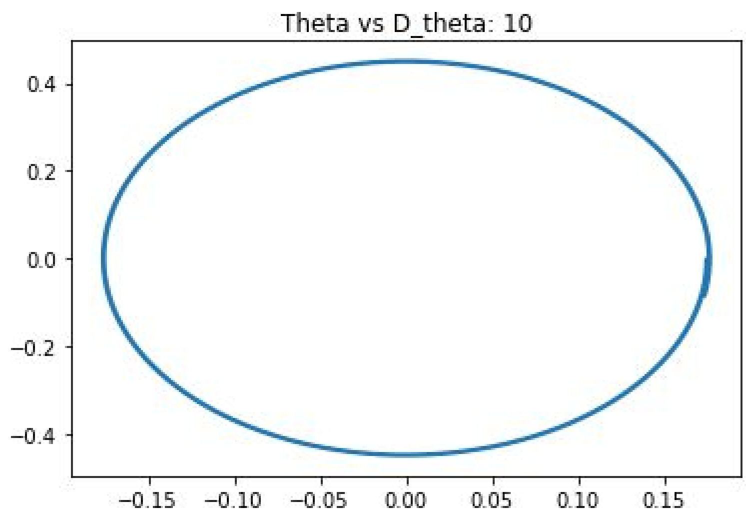
Periods for changing Rod lengths: 2.838453790227457 4.487989505128276 5.310260795610529

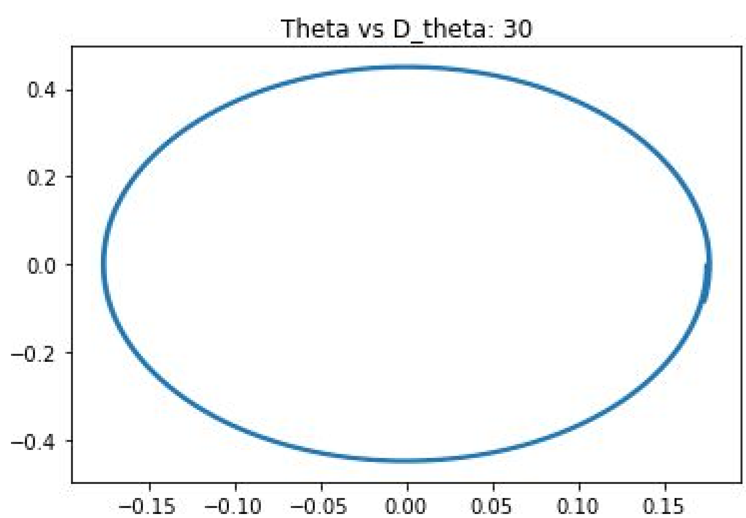
Here we can see that the period changing more with the length of the rod, which is expected. It does not change with the angles up to 45 degrees. This is what we would expect. The longer the length of the physical pendulum, the larger the period. This makes sense if you think of the distant the pendulum has to travel.

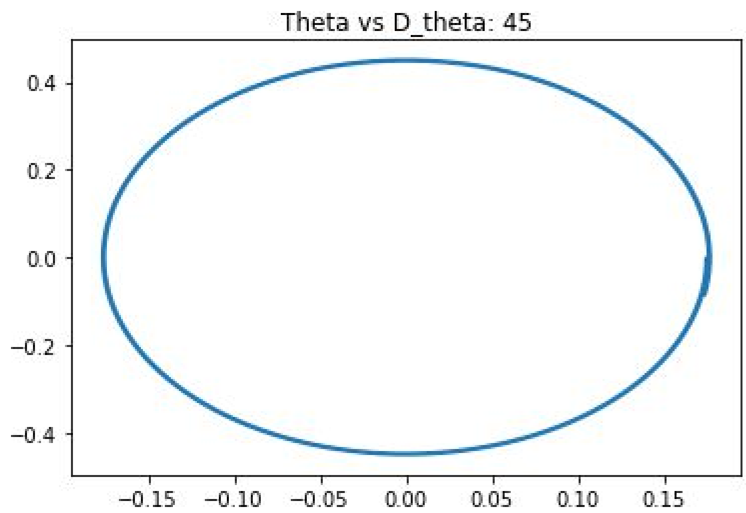
(f) Plot the system in phase-space (i.e., ˙θ vs θ) for a series of different initial angles. Explain the resulting figure and its significance.



Above graph represents one theta. Below is a set of 3 different initial thetas.







Here we see the change in theta vs d_theta. This is what we expect since the two are dependent upon each other and the period does not depend on theta. This graph shows how periodic the two thetas are in regards to their changes over time.

(g) Challenge: Add friction in the form of air drag. How do your figures change?