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
$y(x) = cos(x) + 2 * sin(x)$

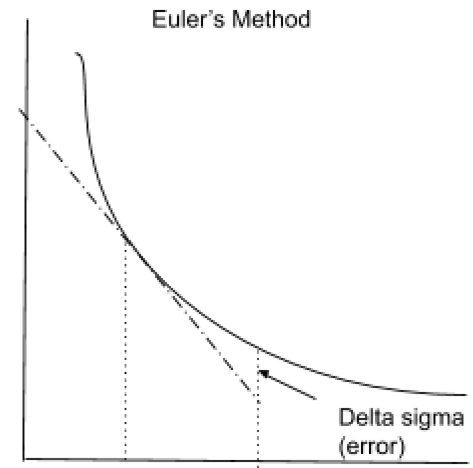
(c) 1st-order, linear differential equation
$\frac{dy}{dx} = 3x + 4$

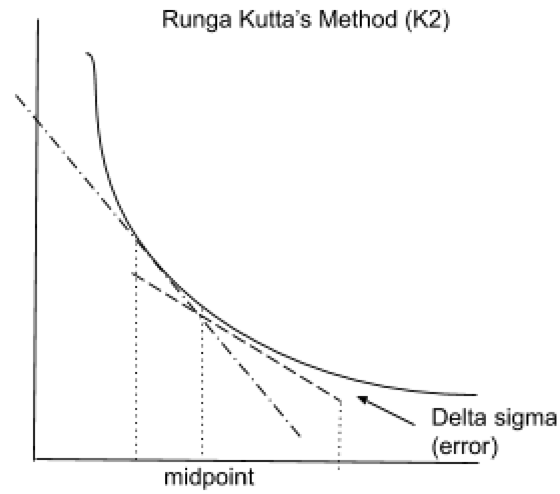
(d) 4th-order, non-linear differential equation
$\frac{d^4y}{dx^4} - x^2 + 2 = 1$

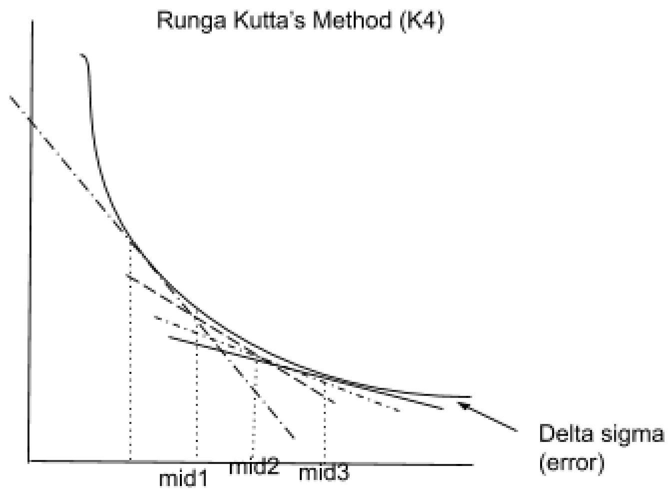
(e) Two coupled, first-order linear differential equations
$\frac{dx_1}{dt} = f(x_1, x_2, t)\ and\ \frac{dx_2}{dt} = g(x_1, x_2, t)$

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.

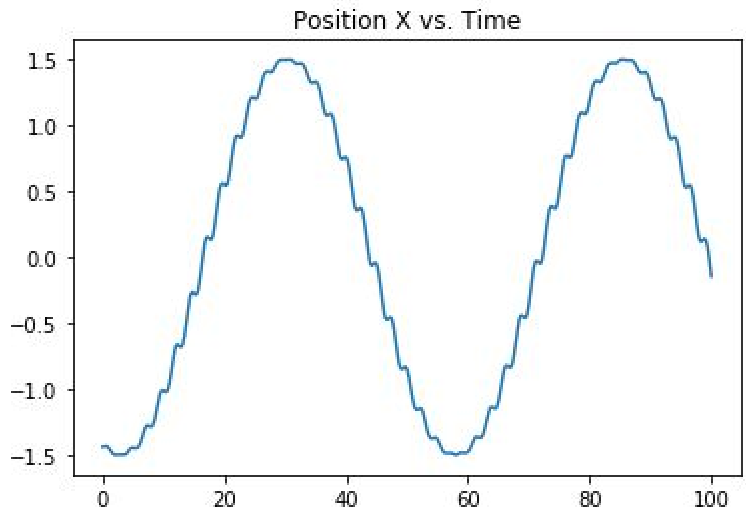
The algorithm I chose to use is the Euler's Method. The error is slightly larger than other potential methods, but this is a more straightforward method for the purpose of this problem, it is okay. This algorithm will allow us to find the position over time when not given any other information to begin.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

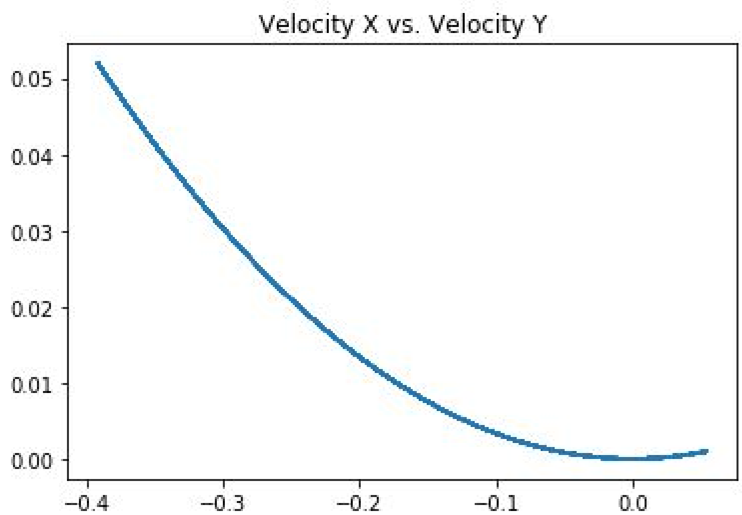
```
Begin:
Import numpy
Import matplotlib.pyplot as plt

#Define variables
G = 9.8
L = 2
A = 2
H = 0.001
Omega = np.sqrt(g/l)
Theta = 0
Theta_dot = 0

#Define all needed arrays:
Theta_array = []
Thetadot_array = []
X_array = []
Y_array = []

#define time and begin for loop
Time = np.arange(0,50,h)
for t in time:
    tdotnew = tdotnew + (omega*A*np.sin(omega*t + theta))*h
    tnew = tnew + tdotnew*h
    period = 2*np.pi*(np.sqrt(l/g))
    theta_array.append(tnew)
    thetadot_array.append(tdotnew)
    x = l*np.sin(tnew)
    y = l - l*np.cos(tnew)
    x_dot = l*np.sin(tdotnew)
    y_dot = l-l*np.cos(tdotnew)
    x_array.append(x)
    y_array.append(y)
    xdot_array.append(x_dot)
    ydot_array.append(y_dot)
Return theta_array, thetadot_array, x_array, y_array, time, x_dot, y_dot, etc.
```

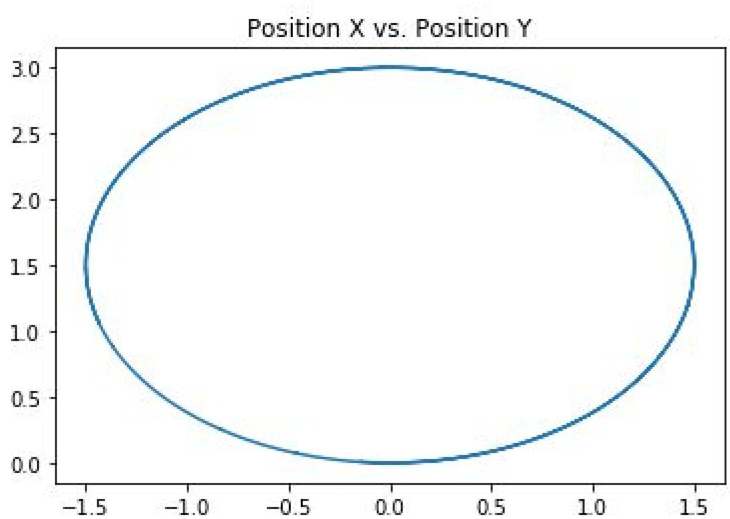
(d) Make a plot of the mass's position in Cartesian space and also plot the y-velocity against the x-velocity. Describe your figures and explain why they make sense.

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



Below is the X Velocity vs Y velocity:





(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?

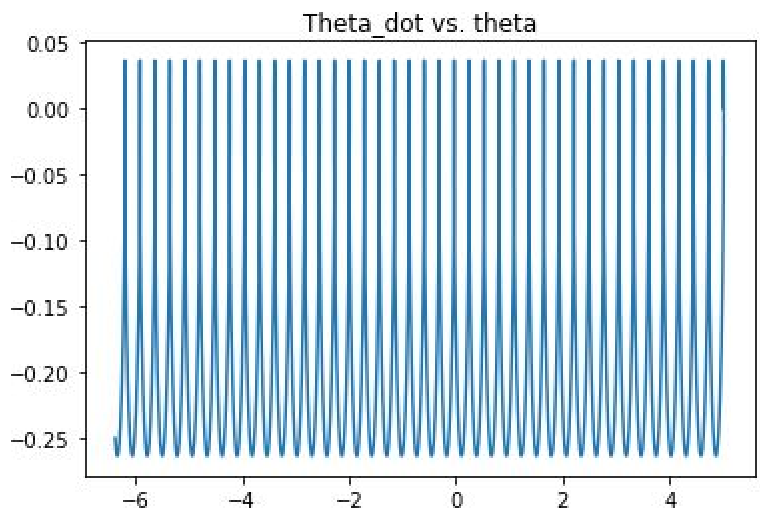
Results of the period for a few different angles and rod lengths:

Periods for changing Thetas: 2.458173089805204 2.458173089805204 2.458173089805204

Periods for changing Rod lengths: 2.007089923154493 2.838453790227457 3.4763817222630955 4.014179846308986

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 that are smaller since we employed the small angle approximation. But, if we have a large enough angle, this will lead to larger results that the theoretical expectation.

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



Here we are seeing the oscillations, the changes in theta over time will be periodic and like harmonic motion.

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