Kernel: Python 3 (system-wide)

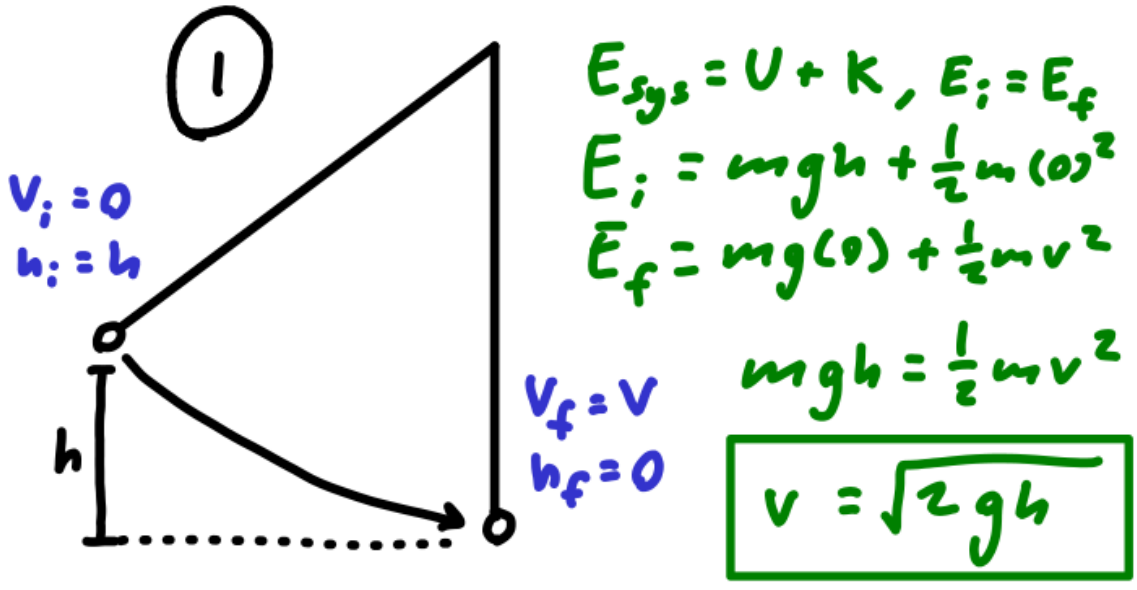
Introduction

The law of conservation of energy states that energy cannot be created, nor destroyed. Across this lab we intend to predict the transformation of energy between its kinetic and potential forms through experiments primarily focused on oscillation. Kinetic energy is the energy of motion meanwhile potential energy is the energy depending on an objects position. There is also potential energy which can be stored in a spring, given by: $E_{sp} = 1/2kx^2$.

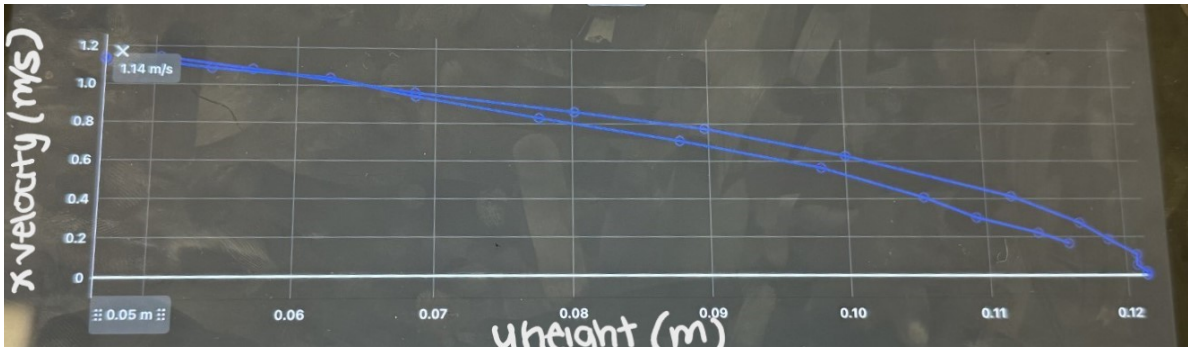
In [1]:
```
# Python imports
from math import *
import numpy as np
import matplotlib.pyplot as plt

g = 9.81 # m/s^2
```

Problem 1: Mass on a String

The calculations are as follows to find the velocity of the ball at the bottom of the arc:



The actual velocity was found using Graphical GW:

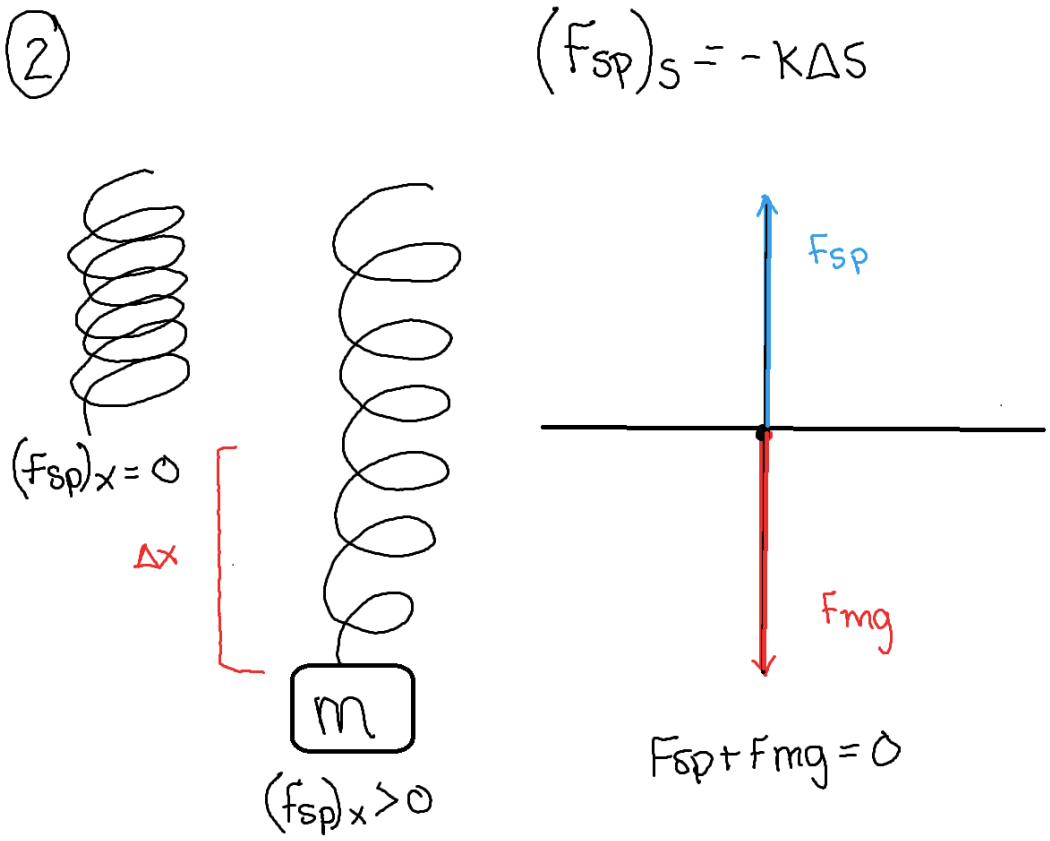


Our measured velocity was fairly close to the calculated velocity. The error can mostly be explained by air resistance and the issue of calculating infinitesimal velocity using discrete methods as was done with video motion analysis.

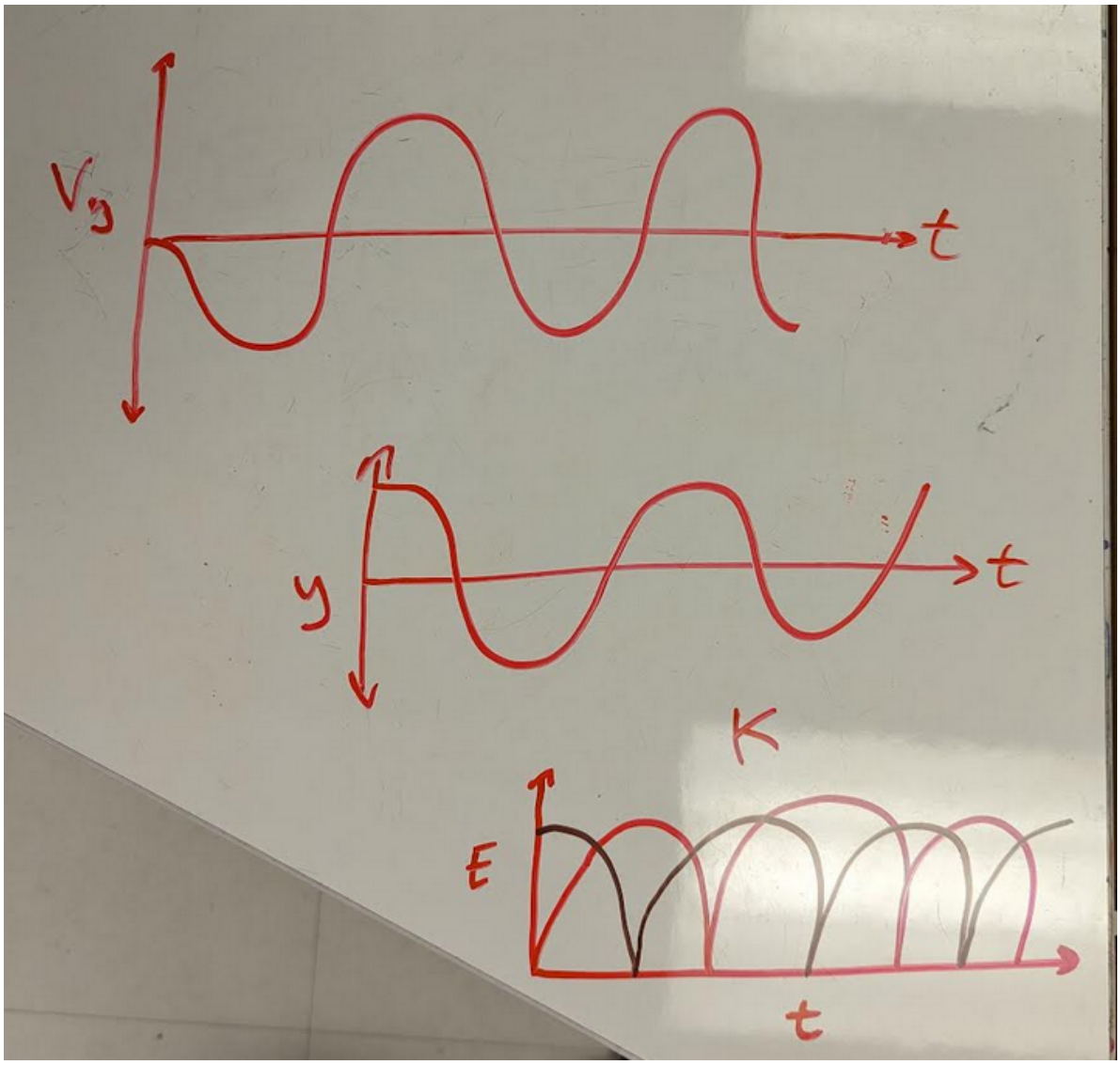
In [12]:
```
init_ball_height = 10 # cm
final_ball_velocity = sqrt(2*(g*100)*init_ball_height) # cm/s
actual_ball_velocity = 114 # cm/s
print(f"Predicted Velocity: {round(final_ball_velocity,3)} cm/s")
print(f"Error: {abs(round(actual_ball_velocity-
final_ball_velocity,3))} cm/s")
```

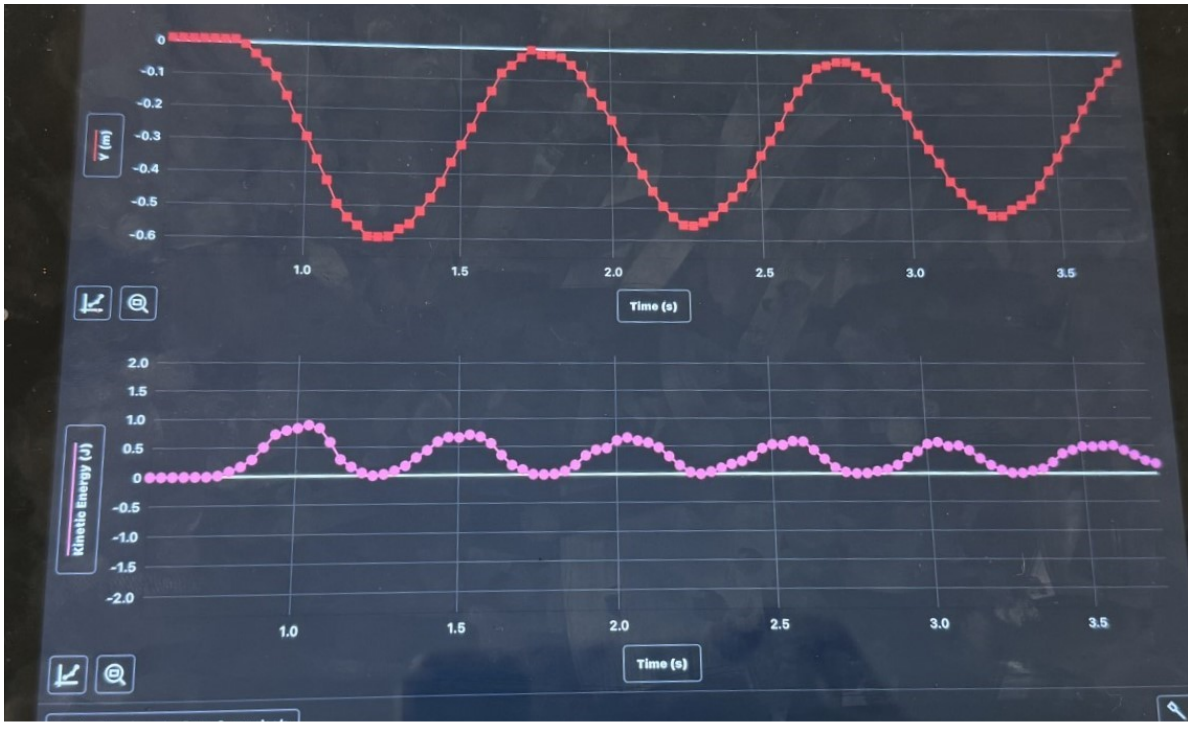
Out[12]:
```
Predicted Velocity: 140.071 cm/s
Error: 26.071 cm/s
```

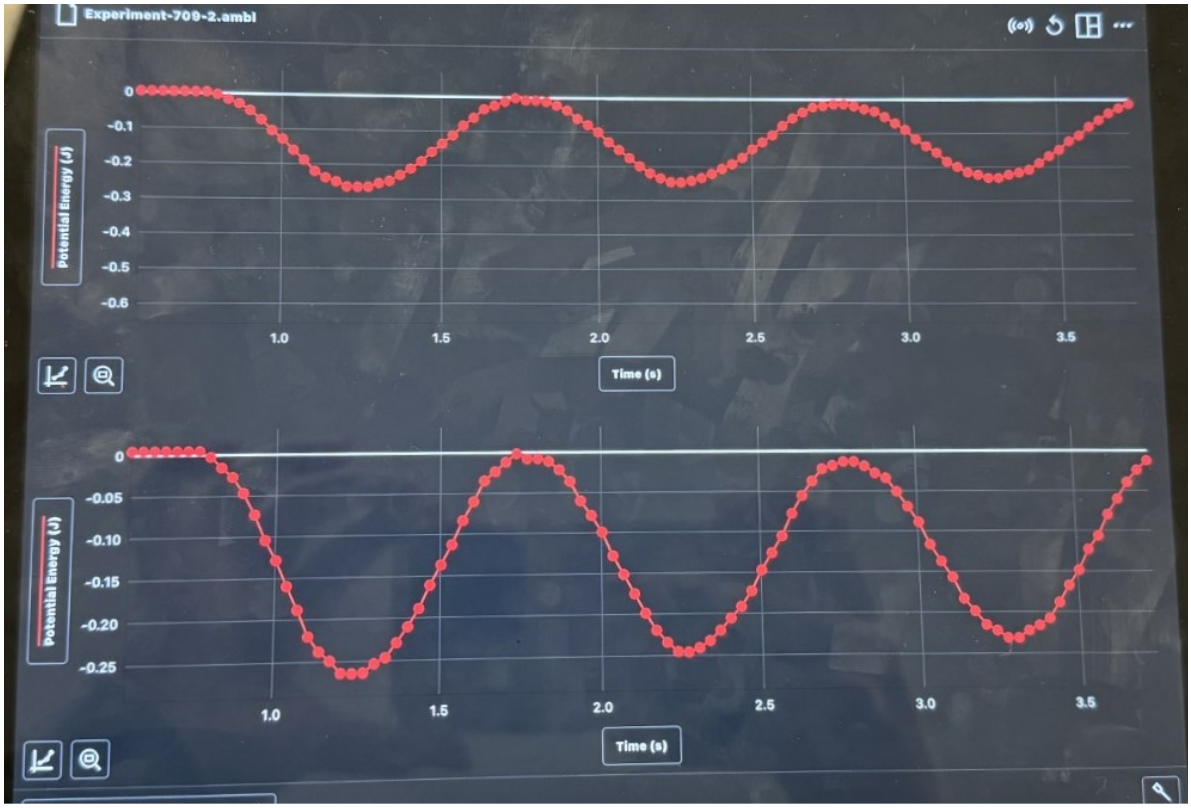
Problem 2: Mass on a spring



Here the force of gravity and the spring are the same, so setting them equal to each other makes solving k easy with $k = \frac{mg}{\Delta x}$.


v_y
t
y
t
K
E
t


0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.6
y (m)
1.0
1.5
2.0
2.5
3.0
3.5
Time (s)
2.0
1.5
1.0
0.5
0
-0.5
-1.0
-1.5
-2.0
Kinetic Energy (J)
1.0
1.5
2.0
2.5
3.0
3.5
Time (s)



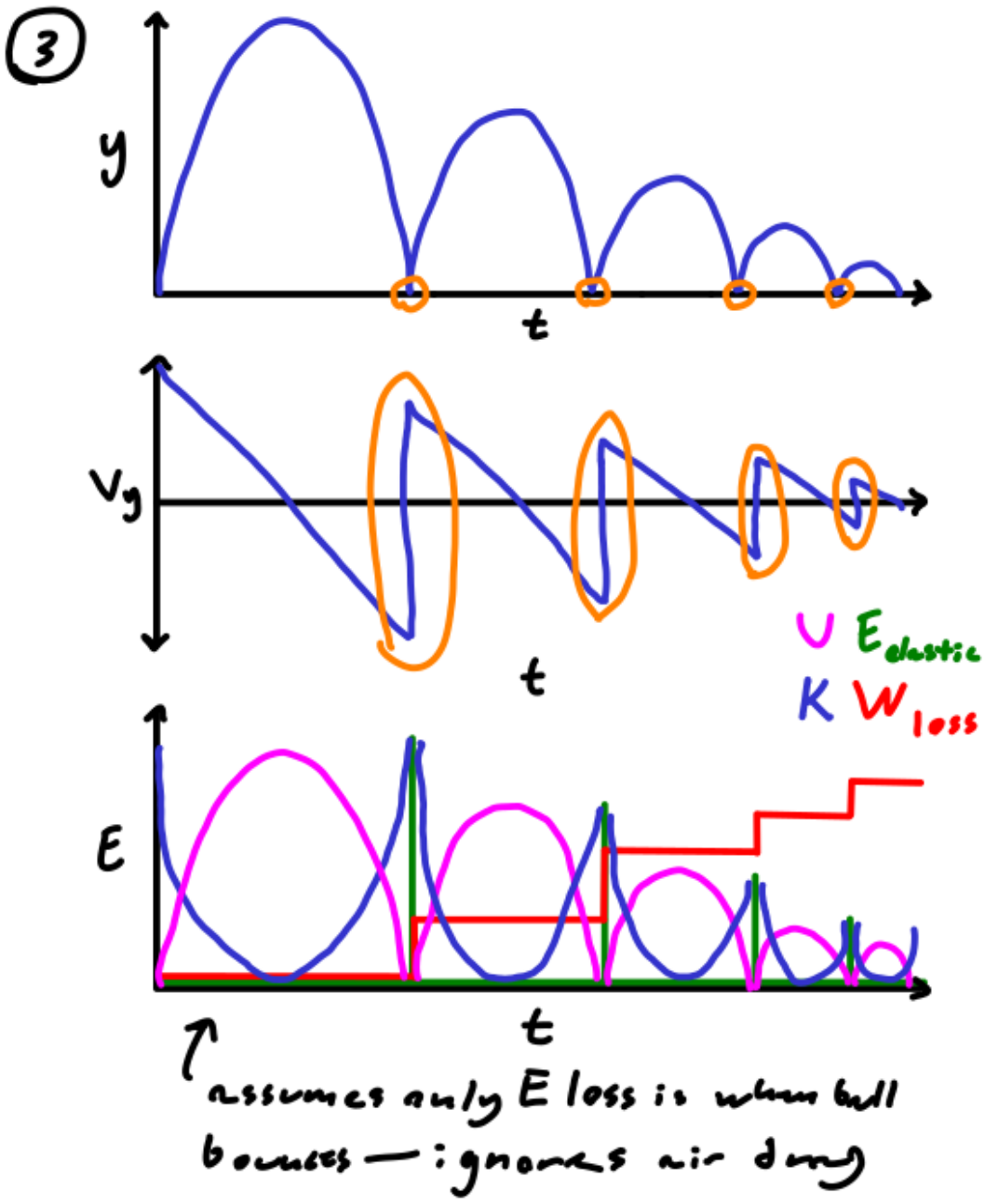
We set the y-origin to be where the ball begins moving down, which is why the y-height is negative, and therefore potential energy. This can be fixed by just translating the graphs up. Overall, though, the graphs are fairly close to our predictions.

In [5]:
```
ball_mass = 0.04592 # kg
non_stretched_length = 26 # cm
stretched_length = 54 # cm
spring_k = (ball_mass*g)/(stretched_length - non_stretched_length)
# N/m
print(f"Spring k: {spring_k} N/m")
```

Out[5]:
```
Spring k: 0.0160884 N/m
```

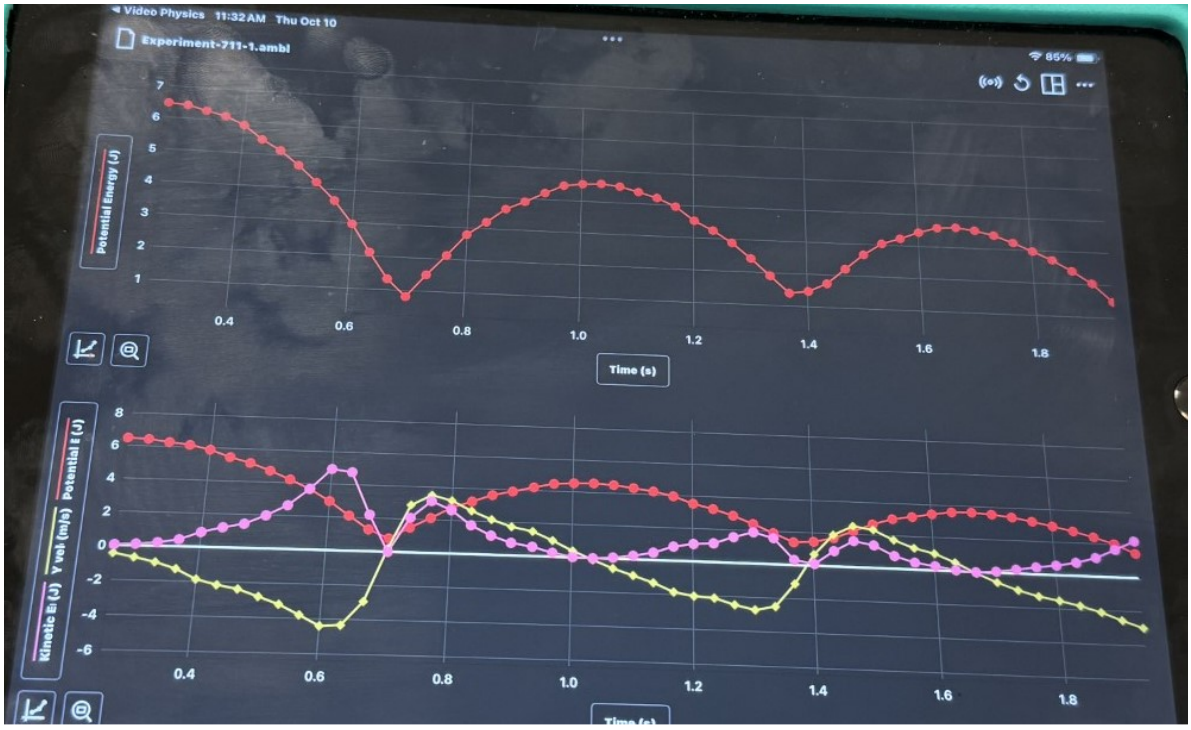
Problem 3: Time to Bounce

Before recording the ball, we made our predictions:



The critical points seem to be where the velocity is zero or where it changes "instantaneously" (when it bounces).

Using the recorded video of the bouncing ball, we got the following:



Our predictions seem to line up with reality here, and the energy graphs are implied from the velocity graph as they were for the predictive graphs.

The total energy of the system is the sum of three energies: kinetic energy, gravitational potential energy, and elastic potential energy. Air drag ("friction") reduces the energy of the system (the ball) by exerting negative work. When the ball hits the ground, the elastic energy conversion from kinetic energy is not one hundred percent efficient, so some energy is lost during those transitions (when the ball bounces).

Conclusion

Our understanding of the conservation of energy was proven correct, within the bounds of human error, for the three problems we did. For the first problem, our actual value was only 26 cm/s less than what we predicted, which can be explained easily as it was due to air resistance and the issue of infinitesimal velocity being recorded discretely by the iPad. The second problem also went well, with the graphs aligning with our predictions, although the axis have different origins. The k value calculated for that problem is also reasonable. The third problem's actual graphs are quite close to the predictions as well, although the real graphs are a bit more messy and less ideal, which makes since due to all the other factors in play.