1.) DECOMPOSE 2ND ORDER ODE

acceleration:

$$m\frac{d^2x}{dt^2} = F_t - F_d - F_r, \qquad \frac{d^2x}{dt^2} = \frac{1}{m}\left[\frac{T}{r_w} - m_w r_w \alpha - \frac{1}{2}C_d\rho A\left(\frac{dx}{dt}\right)^2 - C_r mg\right]$$

$$\frac{dx}{dt} = V_{i+1} = V_i + \frac{1}{m}\left[\frac{T}{r_w} - m_w r_w \alpha_i - \frac{1}{2}C_d\rho A(V_i)^2 - C_r mg\right](t_{i+1} - t_i)$$

$$X(t)_{i+1} = X_i + V_i(t_{i+1} - t_i)$$

deceleration:

$$\frac{d^2x}{dt^2} = \frac{1}{m}\left[-\tfrac{1}{2}C_d\left(\frac{dx}{dt}\right)^2 - C_r mg\right] = a$$

$$\frac{dx}{dt} = V_{i+1} = V_i + \frac{1}{m}\left[-\tfrac{1}{2}C_d\left(\frac{dx}{dt}\right)^2 - C_r mg\right](t_{i+1} - t_i) = V$$

$$X(t)_{i+1} = X_i + V_i(t_{i+1} - t_i) = X$$

Moving_Train

Inputs: Train Parameters

Outputs: Graph of Position & Velocity

Calls: train_motion()
RK4()

RK4

Inputs: Odes, timespan, Initial Values

Outputs: Position/Velocity Data

Calls: Odes

Train_Motion

Inputs: time, position, velocity

Outputs: velocity, acceleration

Calls: N/A

Functions

Operation	Function
Initialize Parameters	#Moving_Train
↓	
Create Odes with train Params	#Moving_Train → Train_Motion
↓	
Approximate Position/Velocity from ODEs	#Moving_Train → RK4
↓	
Graph Position and Velocity	#Moving_Train

Pseudo Code

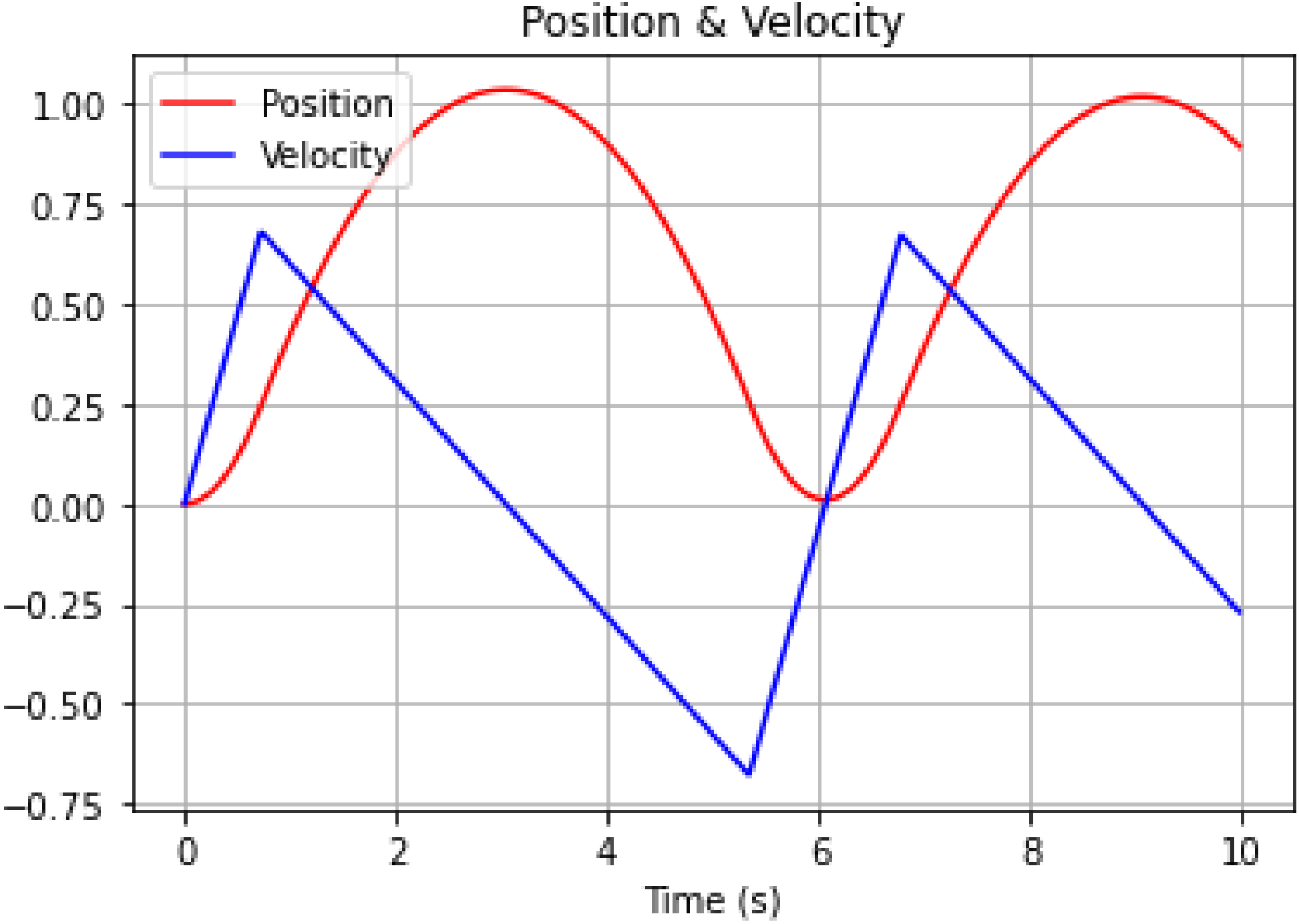
Group Members:
Alex Peters
Nathan Vance
Shawn Haymore

Lab #9 Writeup:

#1: Our function of rk45 converges at a step size of 0.02, we calculated this within the main function, creating different tspans with increasingly smaller step sizes, until we reached a point that caused our approximate error to be less than 0.01.

#2: After a certain number of iterations, the plot does not change drastically, causing us to believe that we have reached a point where increasing the accuracy would only increase computation speed and not have a major effect on our train's performance.

#3: How we chose our step size was using a function within the driver file to iterate increasingly smaller step sizes, and computing the approximate error of the function. Because we have found the appropriate step size for our needs, on future scripts related to this problem we can be more streamlined in choosing a step size, without the need to run our code multiple times. This will help with computation speed.


Position & Velocity
Position
Velocity
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
0
2
4
6
8
10
Time (s)