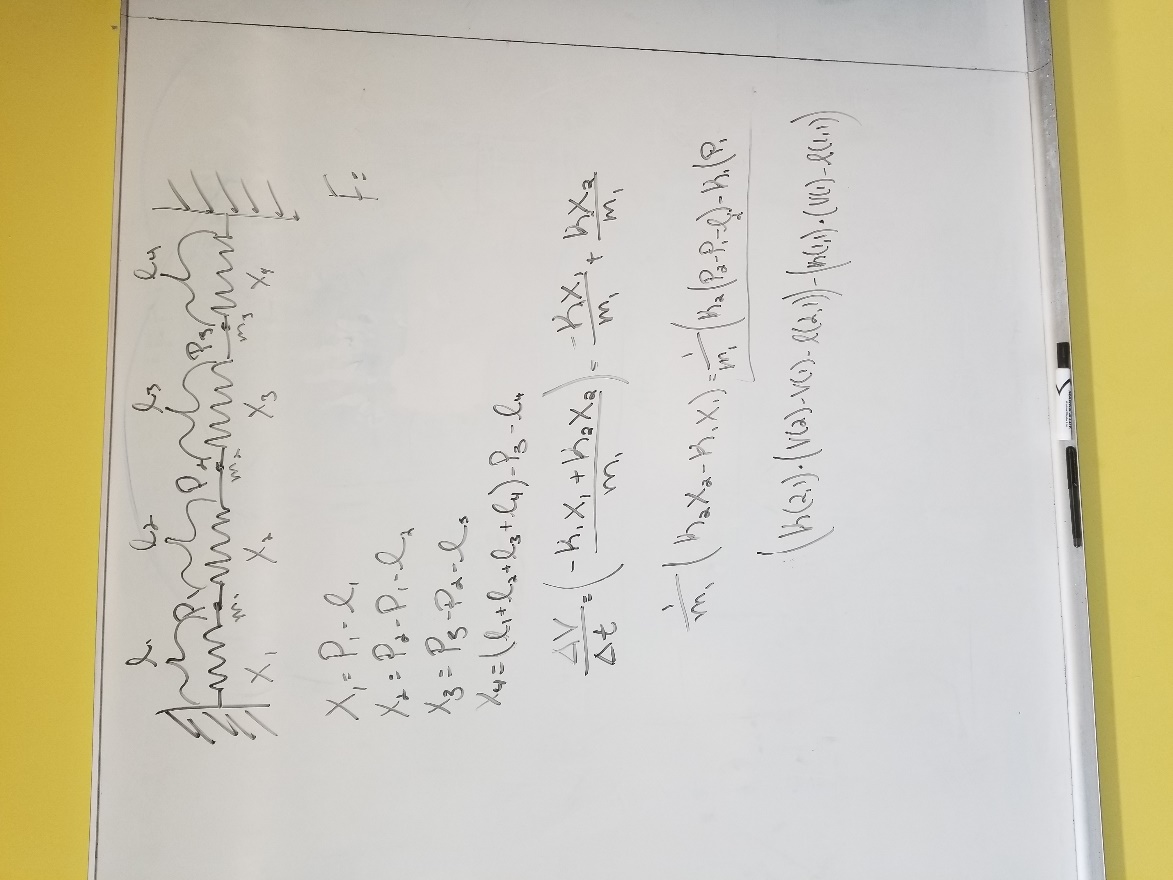
|  |
| --- |
| Introduction to Simulation and Modeling |
| Project 01 |
| Hooke’s Law |

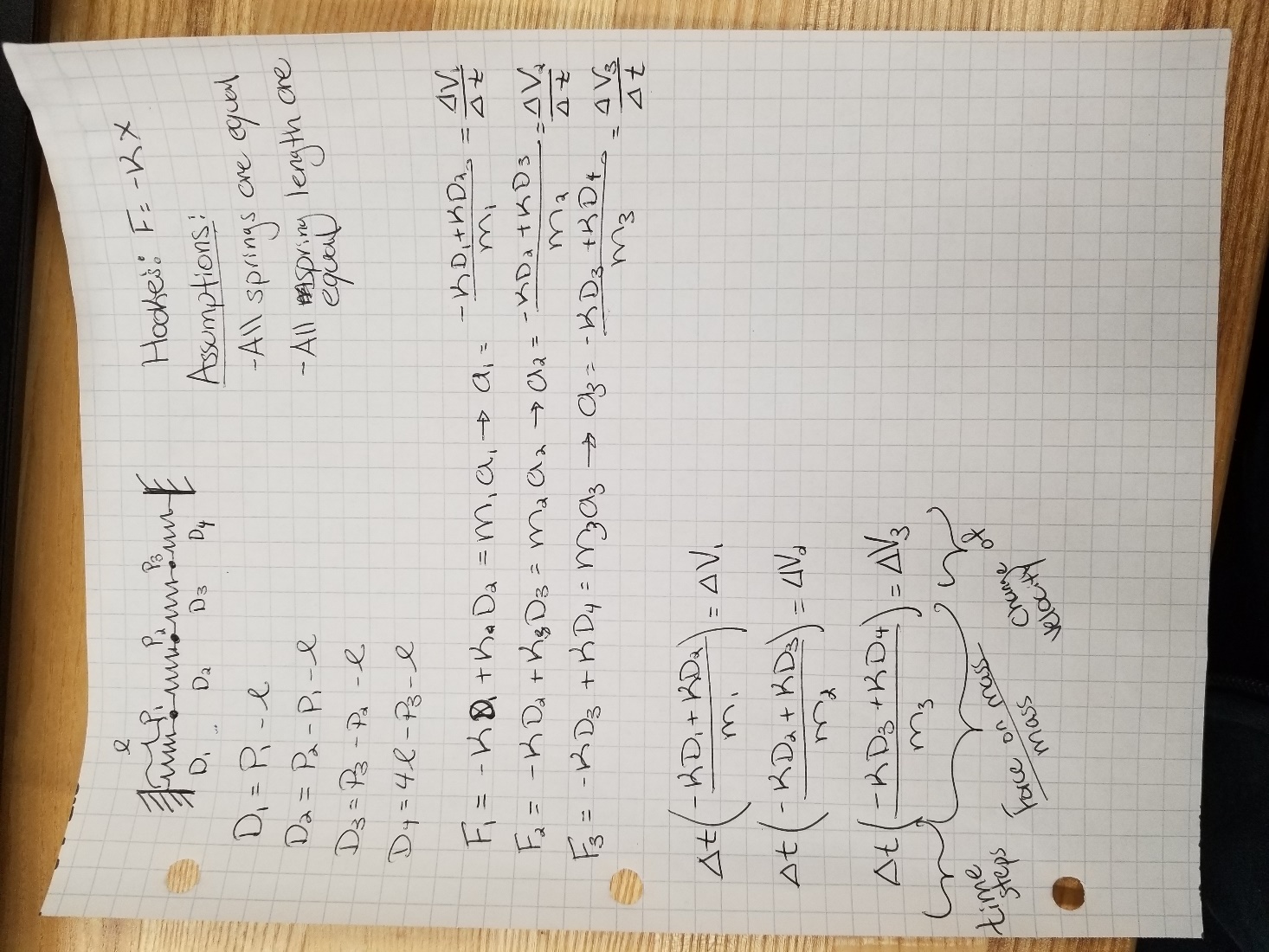
|  |
| --- |
| James Deromedi  2-18-2019 |

I started with writing out the displacement of the springs. And then I wrote out the forces that are acting against the masses, which I assumed as point masses. Then I converted the equations into a MATLAB code. I tried to have each parameter; spring length, spring constant, and mass able to be changed. While I was able to write out the equation on a white board, implementing it in MATLAB did not work. Therefore, I reverted to the original writeup. Below is a photo of the work I completed to be able to have each parameter be changed for only the first mass.

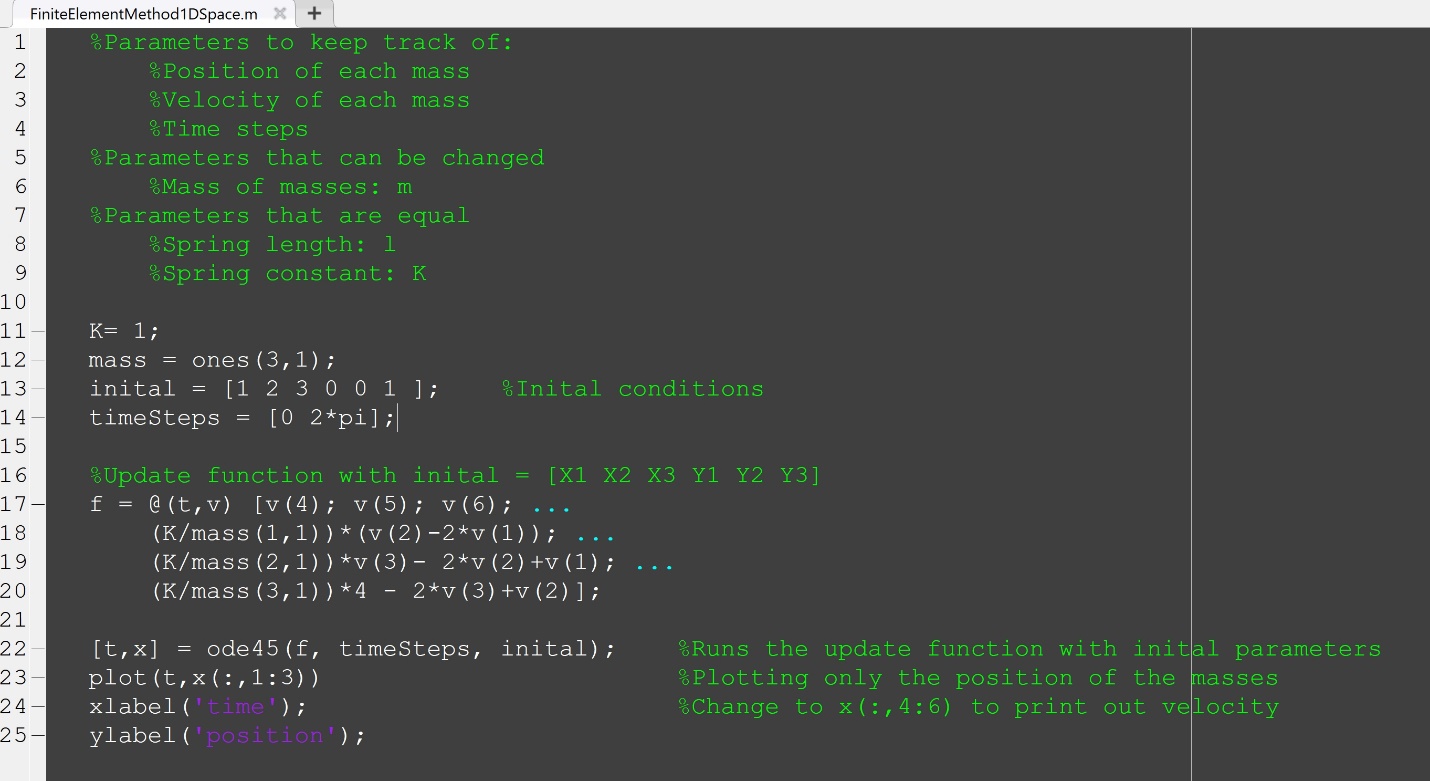
I was not able to get the ODE to reflect off the wall, but logically we would need to send a single pulse through the system to accurately see a reflection off the wall. By increasing the time range, we could see the interference of the reflections by odd dips within the positions of the masses.



1. Collection of parameters and variables that are converted to a 1st order ODE, which can be inputted into MATLAB



1. Implemented ODE into MATLAB



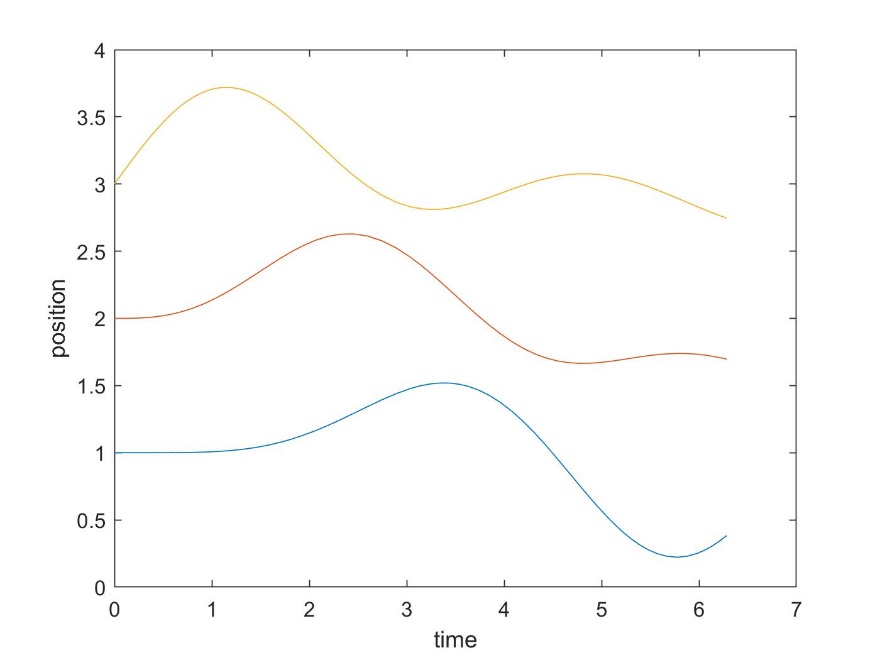
Screenshots of the ODE with different input velocities, same masses, same spring constants, and same spring lengths

Figure 1: Initial parameters [1 2 3 0 0 1]

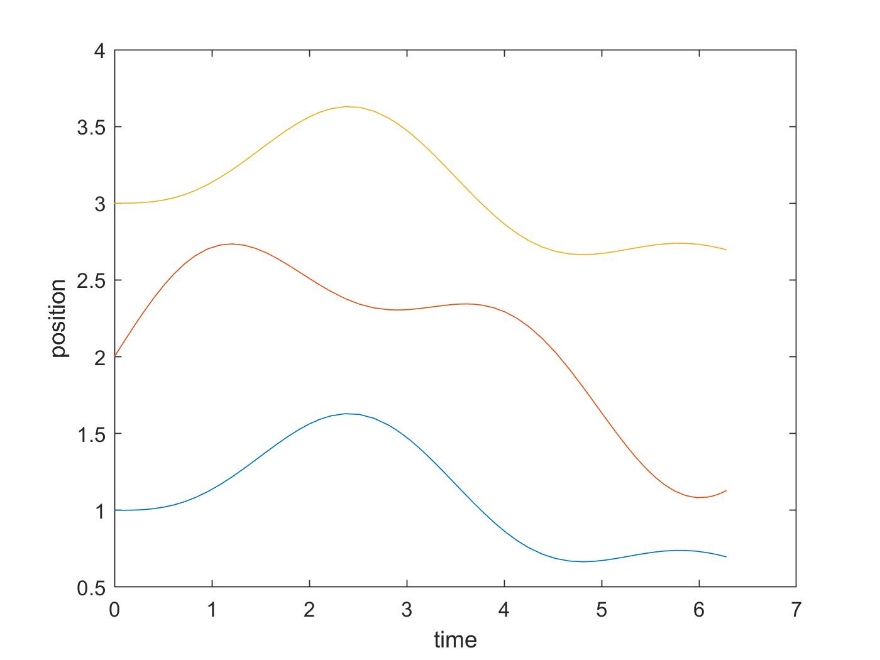
1. 

Figure 2: Initial parameters [1 2 3 0 1 0]

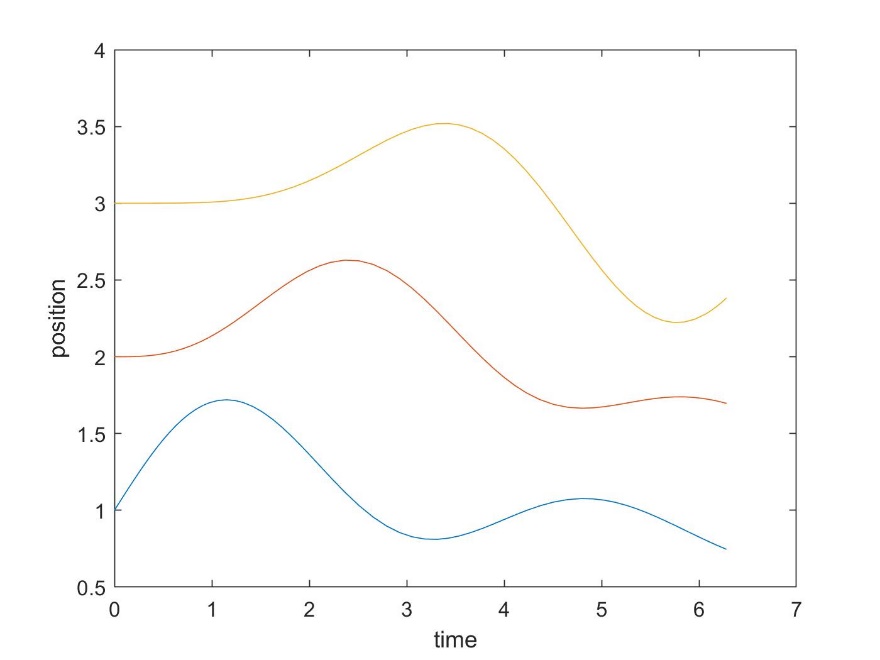
1. 

Figure 3: Initial Parameters [1 2 3 1 0 0]

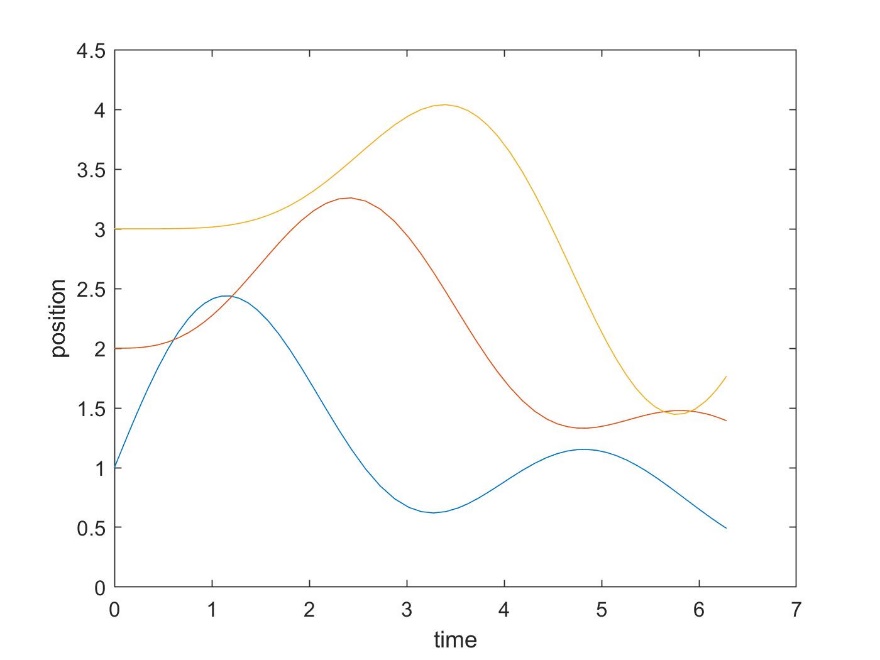


Figure : Initial parameters [1 2 3 2 0 0]