

ECGR 3120 Group Project

Sylvester Pudelko

November 10, 2025

Introduction

The goal of this project was to design an ink jet printer simulation. In this printer, it would use a charged particle, with a charge of -1.9×10^{-9} C, which would then pass through a set of parallel plates. Depending on what voltage these parallel plates were, it would affect the trajectory of where the ink would land. By exploiting this, you would be able to form any letter you would like, and at any resolution given. In this case, a resolution of 300 DPI was chosen.

To be able to simulate this, it was chosen to write this simulation in Jupyter notebook using python. By doing this, you are able to run the individual sections that are wanted. To be able to show the animations and graphs, the Python library Matplotlib was chosen. By utilizing its 2D graphing ability to display information, and its 3D graphing ability to give animation, I was able create an accurate simulation on how a charge ink droplet would behave when passed through a parallel plate with an electric field. Additionally, I was able to time all the results in simulation, and verify through hand calculations that these values are theoretically correct.

Below are a couple of links related to the project:

- [Link to the YouTube video demonstration](#)
- [Link to the GitHub containing scripts and animation videos](#)

Part 1

For part 1 of the project, the goal was to create a simulation of a charged particle passing between the parallel plates when there is no electric field in between. According to hand calculations, $\frac{.003M}{20M/S} = 0.00015$ seconds. When the simulation was run, the time it took for the the droplet to reach the piece of paper was also found to be 0.015. A video and GIF animation of the trajectory and calculations can be found in the GitHub. Below is a photo of the result, where there is a dot right in the middle of the paper.

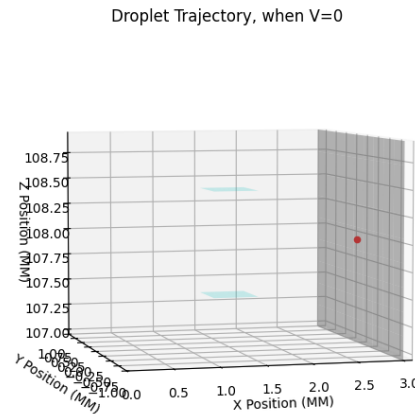


Figure 1: End Result of Part 1 Animation

Part 2

For part 2 of the project, the goal was to draw the letter 'I' at 300DPI. The task was also to time how long it would take. To accomplish this, a voltage of 2360 was used, as this was the maximum voltage found to correctly simulate before the droplet hit a parallel plate. From this, the I was drawn to be 5.34mm long. We can then calculate how many dots we need to get a resolution of 300DPI. We can convert it to DPMM, which was about 12. From this, it was found that we needed about 60 droplets to achieve this resolution. Additionally, to be able to draw the dots at separate spots, the voltage needed to be changed by 80 Volts each time that a droplet finished its trajectory. From this, we were able to create a 300DPI letter 'I'. The calculated time that it would take to accomplish this task was 0.009 seconds. Below is a final result from the animation. A video and GIF animation are both found on GitHub.

Droplet trajectories when the voltage is varying

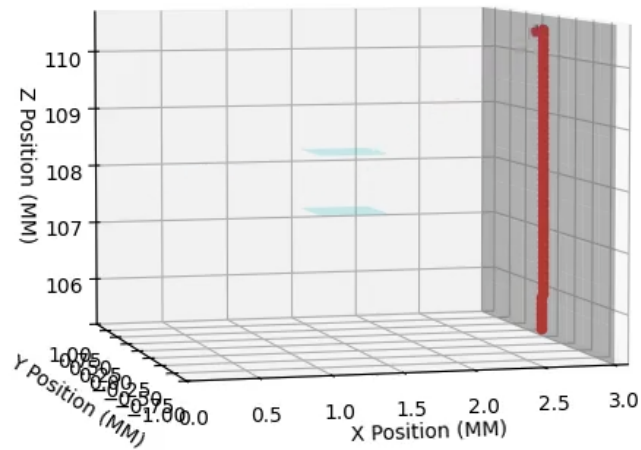


Figure 2: End Result of Part 2 Animation

Part 3

In part 3, the goal was similar to part 2, where we had to draw the letter 'I' at 300DPI. However, this time, the goal was to draw it as large as possible. Another task was to have a graph of voltage versus time. A third item requested was to speed up the time it takes for the letter to print. For the first part, the same voltage was used as before, as going above 2360 volts would cause the droplet to hit one of the plates. So the maximum size found was 5.36mm. For the second part, a graph of the voltage versus time was found by taking the voltage measurements alongside a time measurement. From there, you are able to graph the data, and it should be shown as a staircase function. For the third part, to be able to speed up how quickly the item prints, a new strategy was adopted regarding how ink is dispensed. Instead of the ink being sent out once the previous droplet has reached the end, once the droplet has reached the midpoint, then the next drop of ink is then shot out. By doing this, we were able to cut down the amount of time it took to print the letter 'I' by half, so our print time decreased to 0.0045 seconds. Below is the final result of the animation. A video and GIF animation are both found in the GitHub repository.

Droplet trajectories when the voltage is varying

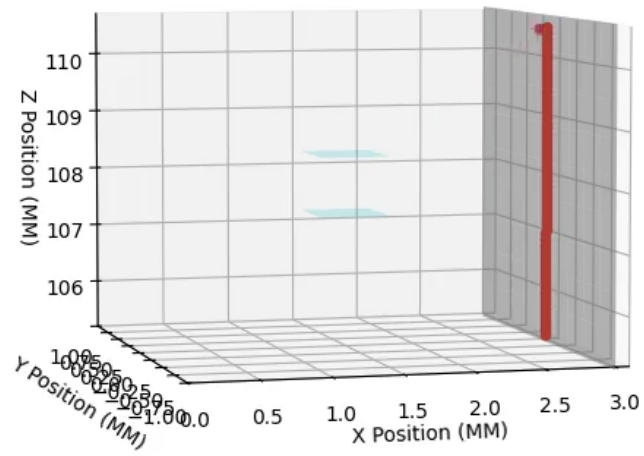


Figure 3: End Result of Part 3 Animation

Below is the graph of the voltage versus time.

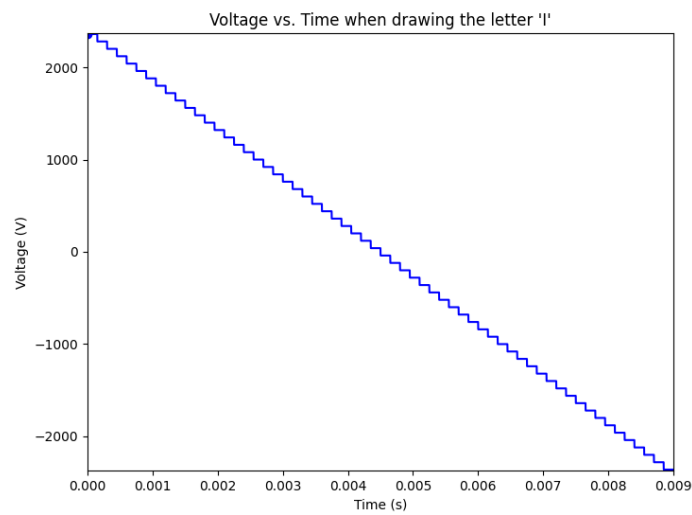


Figure 4: Voltage versus time

As expected, the graph appears to be a staircase function.