Lab 4 Report

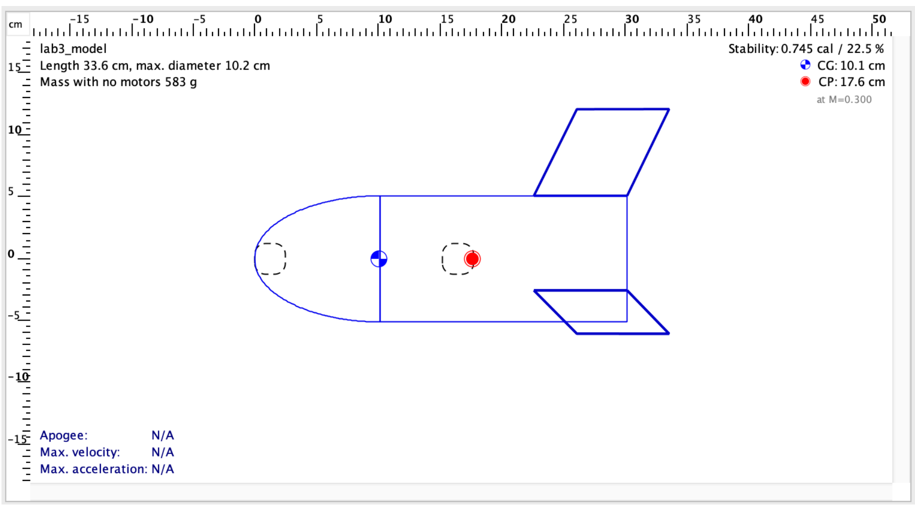
Ashna Khemani, Katie Li, Christine Meng, Hannah Youssef

I - Methods and Materials

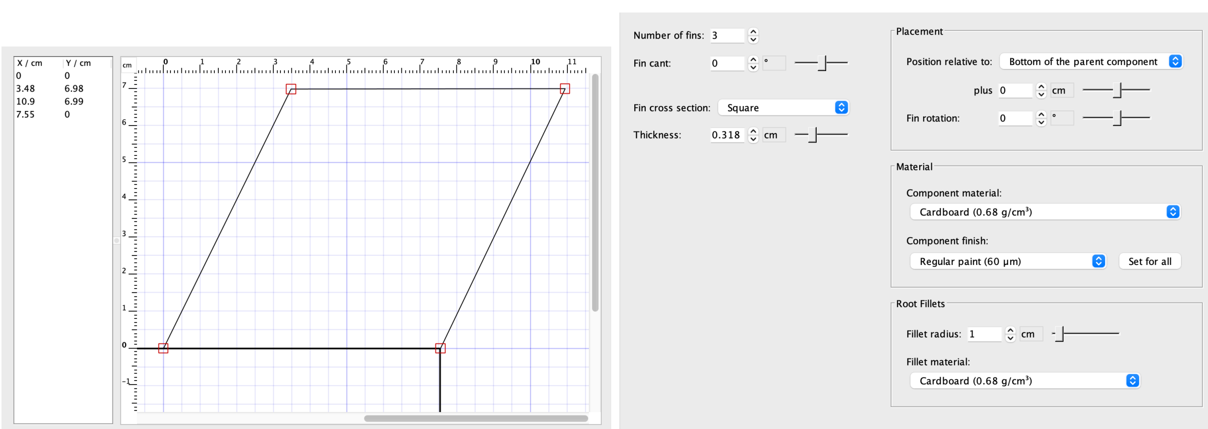
Designing with OpenRocket

We believed having a well-designed and stable rocket would help keep our performance predictable. We used OpenRocket for basic calculations, and to change our bottle rocket configuration to get a good stability number.

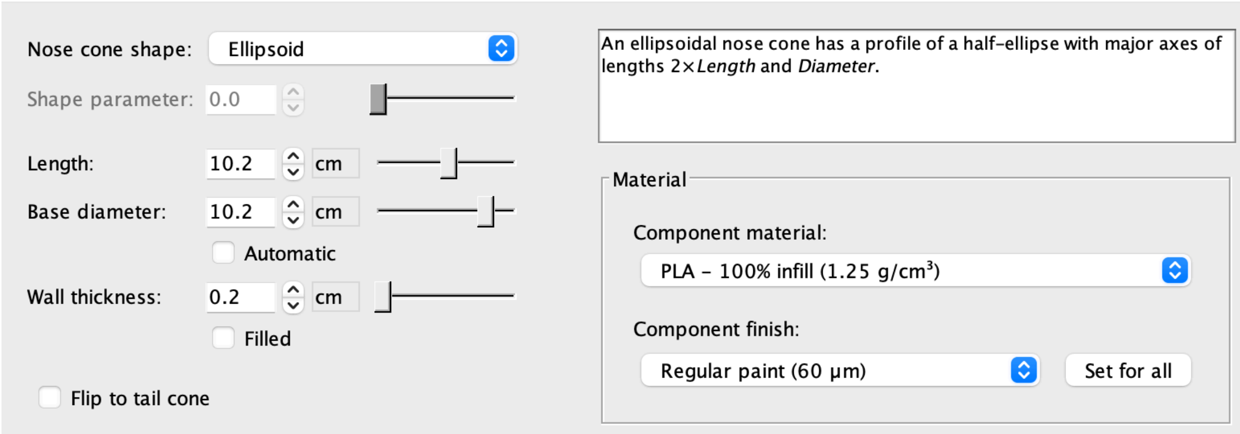
Ideally, we would've wanted stability between 1~2cal, but settled for 0.745cal (shown in top right of below diagram) since we were flying for relatively short distances. Also, it would be harder to get a conventionally good stability for a short and stubby rocket. We primarily altered the nosecone shape and mass, and fin shape. A main goal was to get the Center of Pressure well behind the Center of Gravity, which we achieved.



Fig(1): OpenRocket model. CG and CP locations shown. Precise locations are written in top right corner. (Stability=0.745cal, CG=10.1cm, CP=17.6cm)



Fig(2): Fin configuration.



Fig(3): Nosecone configuration.

OpenRocket was also able to give us a breakdown of the drag coefficients. For our calculations, we will use the Total Cd for the full rocket (0.309):

Component	Pressure C_D	Base C_D	Friction C_D	Total C_D
Total (Rocket)	**0.084 (0%)**	**0.132 (0%)**	**0.093 (0%)**	**0.309 (0%)**
Nose Cone	0.019 (0%)	0 (0%)	0.021 (0%)	**0.04 (0%)**
bottle	0 (0%)	0.132 (0%)	0.048 (0%)	**0.18 (0%)**
Freeform Fin Set	0.022 (0%)	0 (0%)	0.008 (0%)	**0.03 (0%)**

Fig(4): Drag coefficients. 0.309 is the total Cd which we will use in our model.

Construction

We initially laser cut our fins from ⅛'' acrylic for Lab 3. However, they snapped off after our first launch, so we switched to tape-covered cardboard, which was more flexible to prevent breakage when landing and lighter to reduce the weight. Our nosecone was 3D printed at the RPL. Fins and the nosecone were lightly sanded and hot glued to the bottle. We also covered our nosecone with duct tape so it could be protected/somewhat held together if it broke on landing.

Experiments and Data

In lab 3, we primarily ran tests that varied the water temperature while keeping constants of 600mL of water and 20g of butane. However, during the lab 3 demo and later tests, we found increasing the water amount to 750~1000mL gave a more consistent launch, as the 600mL would burn and eject too quickly.

We first tested with relatively high temperatures, of about 35~40°C. This resulted in a quick burn, which led to the rocket barely getting off the launch rail.

We adjusted our tests accordingly to increase water volume and lower temperatures. With **1000mL of water at 18°C** and **20g of butane**, our rocket nearly went over the Quain Courtyard wall, traveling about **55 feet**. We tested the same specs again and found that the rocket launched consistently well. We also adjusted our method of launch to holding the nozzle rather than quickly swinging the bottle; closing the nozzle allowed pressure to build and prevented the fuel from bursting before the launch angle was set.

We decided that our strategy could be to vary the amount of butane while keeping all else constant. We did not run too many tests out of fear of breaking our nosecone right before the final demo, as it was already starting to scrape and dent in some places.

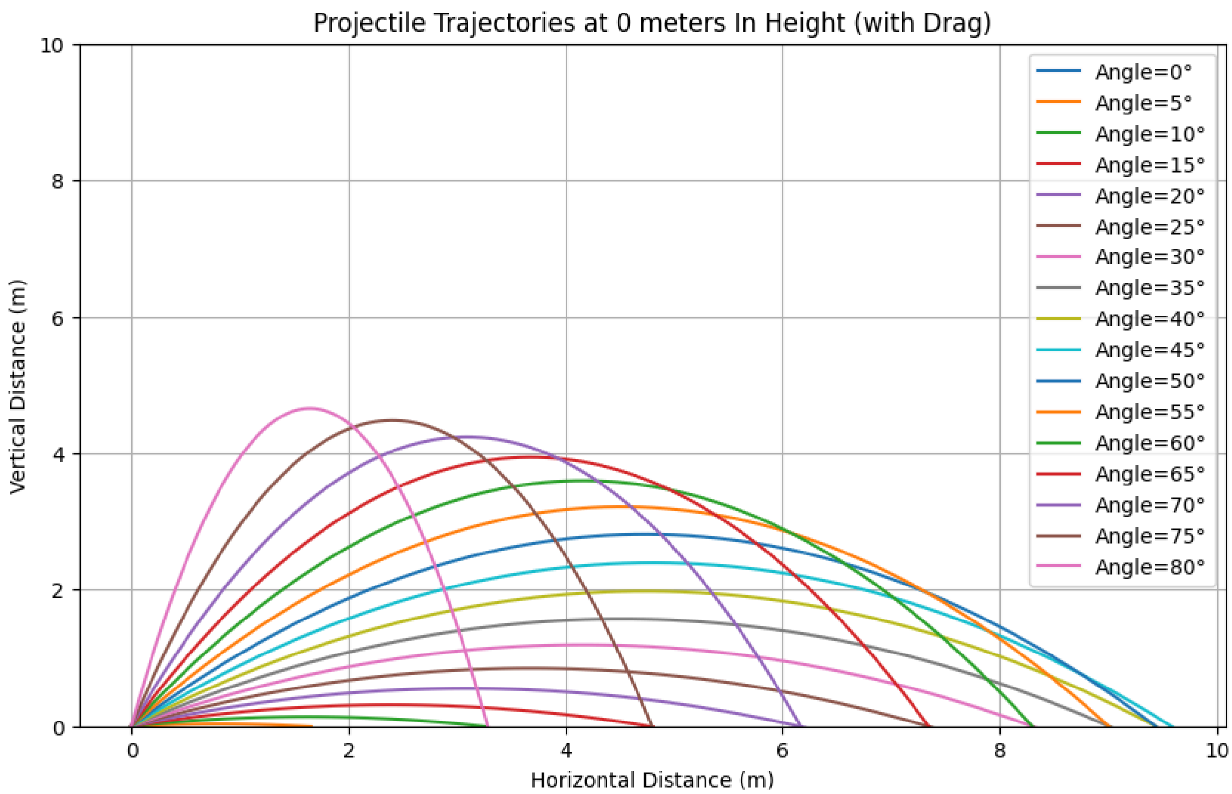
II- Model

We were unable to properly record sensor data, so our calculations were derived from motion tracking our parallel videos to see initial velocity and an accurate angle.

```
import numpy as np
import matplotlib.pyplot as plt

def projectile_trajectory(height, angle, initial_velocity,
drag_coefficient):
    g = 9.81  # gravity (m/s^2)
    rho = 1.225  # density of air (kg/m^3)
    A = 0.01  # cross-sectional area (m^2)
    m = 1.12 # in kg

    angle_rad = np.radians(angle)
    v_x = initial_velocity * np.cos(angle_rad)
    v_y = initial_velocity * np.sin(angle_rad)
    x = 0
    y = height
    x_traj = [x]
    y_traj = [y]
    dt = 0.01  # seconds


    # calculating drag
    while y >= 0:
        v = np.sqrt(v_x**2 + v_y**2)
        F_drag_x = -0.5 * rho * v**2 * drag_coefficient * A * v_x / v
        F_drag_y = -0.5 * rho * v**2 * drag_coefficient * A * v_y / v

        # velocity components (force over mass is acceleration, )
        v_x += ((F_drag_x)/m) * dt
        v_y += ((F_drag_y - g)/m) * dt
```

```
        # position components
        x += v_x * dt
        y += v_y * dt

        # trajectory points
        x_traj.append(x)
        y_traj.append(y)

    max_height_index = np.argmax(y_traj)
    max_height = y_traj[max_height_index]

    return x_traj, y_traj, max_height

# initial parameters
initial_velocity = 9.3
height = 0  # initial length launched (m)
drag_coefficient = 0.309  # coefficient of drag

# Angles from 0 to 80 degrees, with a new graph every 5 degrees
# note: you can change angles too but anything above 80 degrees is
unrealistic
# Plot trajectories for angles from 0 to 30 degrees, with a new graph
every 2 degrees
angles = np.arange(0, 81, 5)

# plotting a graph :3 YIPPEE YIPPEE YIPPEE YIPPEE YIPPEE
plt.figure(figsize=(10, 6))
for angle in angles:
    x, y, max_height = projectile_trajectory(height, angle,
initial_velocity, drag_coefficient)
    plt.plot(x, y, label=f'Angle={angle}°')

plt.title(f'Projectile Trajectories at {height} meters In Height (with
Drag)')
```

```
plt.xlabel('Horizontal Distance (m)')
plt.ylabel('Vertical Distance (m)')
plt.ylim(0, 10)  # Set y-axis limit to show trajectory clearly
plt.legend()
plt.grid(True)
plt.show()
```



Fig(5). Calculated trajectories for the demo day launch with different starting angles. We chose 60 degrees as it corresponds with the target distance of 8 meters.

III- Results

Our performance did not exactly match our calculated trajectory from the model. We launched at a 60 degree angle with 1000 mL of water at 35 C. We had an actual distance of 6.1m, so we were short by 23.75%. Our final score was 114.3.

However, the mismatch with the model may be attributed to the imperfect launch method on demo day. As the rocket swung upwards instead of properly following the launch rail, the actual angle of the rocket was inaccurate to our model and greater than 60 degrees; this increases the vertical distance and reduces the horizontal distance traveled. The rocket was also not aligned properly with the rail, skewing the trajectory (as indicated by the rocket landing at the top of the stairs rather than the end of the marked path). This all contributes to our undershot, as well as why our rocket missed the target hole.

Based on the force generated by the rocket as seen in the video, a steadier launch likely would have resulted in a trajectory accurate to our model. A larger mass (such as a thicker nose cone or slightly heavier fins) would help balance and stabilize the rocket on launch. We would need to adjust the calculations in the model to account for this change in mass, but it should otherwise be effective and functional.

That said, our model could be further refined with telemetry sensor data to verify the starting velocities and acceleration we calculated. The telemetry devices are at times fickle and unreliable, so we would use both the sensor data and the motion tracking through recorded video to determine the model.

IV - Discussion

Consistency:

At the end of lab 4, we can finally reflect on the results of our tests and performance on demo day to define the important parameters that affect consistency and predictability of launches. We concluded that the important parameters were:

1. **Orientation on rail system:** rocket's placement on the rail affected how well the launch would be significantly. On many of the tests we performed, the rocket would lose contact with the rail seconds after release, and that fact made the predictions from the model vary from the actual results. That the rocket would lift immediately after launch also implies that a **heavier rocket mass** may help with consistent launch trajectory.

While the rail system was the most important parameter, here are other parameters that were also significant:

2. **Launching mechanism:** Many inconsistencies stemmed from human errors in pouring the butane into the bottle at the right angle and speed, as well as the way the bottle would be flipped for launch. Waiting a relatively long time after pouring the butane most likely ends up with a failed launch, as the butane begins to boil. Pouring the butane fast and directly on the water causes more water to mix too early with butane and also results in failure. Releasing the rocket while the expansion of butane is still happening causes "sloshing" of water around the bottle which affects COM and makes the launch less predictable. The more consistent the launching mechanism, the more correlation we found between changes in the temperature, volumes and rocket kinematics.
3. **Nosecone and fins:** Adding a massive nosecone made the trajectory of the rocket more predictable. On demoday, the rocket contacted the ground exactly at the tip of the nosecone (which made it break) but also it made drag's effect less significant.

Therefore, to improve consistency, we suggest:

1. Releasing the rocket at the end of the rail so that it doesn't stay in contact for a long time while maintaining the angle.
2. Releasing the rocket a few seconds after the pressure builds up (blocking the nozzle) to avoid sloshing.
3. Releasing the rocket exactly in the middle of the rail, meaning the person who is launching shouldn't have an orientation to the left or right before releasing the rocket (or has another method of ensuring the rocket is straight on the rail).
4. Add a larger nose cone and fins to the rocket, and use online software to figure out where to place them to change the COM and COP.
5. Adding mass to the rocket, which could be just making the nose cone thick (reduces drag) and prevents the rocket from lifting off the rail.
6. Gradually pour the butane at an angle so it slides on the walls of the bottle. This way, a layer of butane forms slowly on the water surface. The thicker that layer, the less butane that is boiling from contact with the water, and thus the further the launch once flipped.

Demoday Performance:

On demoday, we launched at a 60 degree angle with 1000 mL of water at 35 C. Our final score was 114.3. The rocket launched to the side and landed on the steps in Quain courtyard, so it didn't pass through the hole or land on the target, but the perpendicular distance was very close to the hole. It landed on the tip of the nosecone.

Looking back at the video, it looks like the issue was with the launch orientation. The rocket was released slightly skewed to the right, and that made it move to the right instead. If it was to be launched straight forward, it would have passed the hole or slightly move further.

To improve the launch, we could make the orientation exactly in the middle, and use a less steep angle at 50 degrees, this would make the rocket's range longer while keeping the maximum height less high.

Link to final launch video:

https://drive.google.com/file/d/11cFGb9tX4QnMHMDqGy6ofPd0fI2Jzx1Z/view?usp=sharing

Our expected model performance was that the rocket would pass through the hole, but since the rocket was skewed to the right, we didn't match our model. If we were to eliminate launching errors, we would have reached a closer result to the model.