AA103 Final Project

Jonny Dyer

May 22, 2019

Introduction

For our final project, you will be designing, building and launching a water rocket! Water rockets are generally made from 2 liter soda bottles filled partially with water and pressurized with air. The water is exhausted through a nozzle creating thrust and launching the bottle into the sky. There is a vibrant group of garage builders on the internet and some have achieved amazing performance, flying rockets to over 2000 feet in altitude! (albeit not with soda bottles...)

While without smoke and fire, water rockets are a lot of fun. This project is broken into two parts:

1. Pre-work problem set covering the analysis and design of the rocket
2. The rocket build and altitude contest

Rules and Constraints

Since we will be holding a contest, there are some rules and constraints that must be followed:

1. Everyone must use a standard 2 liter soda bottle. The beverage choice is up to you...
2. All rockets will be pressurized to 7 bar (7 × 10^5 Pa)
3. All rockets will carry a 10 g altimeter payload on top
4. All rockets must use the provided bottle cap / nozzle. However, the nozzle internal hole may be modified by the teams

Pre-work - Analysis and Design

Problem 1 - ΔV Analysis

First we will ask you to compute the ΔV your rocket is capable of delivering. A water rocket is referred to as a "blow-down" system as there is no external reservoir of pressurant maintaining constant pressure in the rocket. So the pressure inside will decay as water is expelled.

In general terms, a gas expansion process can be represented as

$$pV^n = \textbf{const}$$

a) **[Question]** What is *n* for the three following processes?
   i. An *Isobaric* (constant pressure) process
   ii. An *Isothermal* (constant temperature) process
   iii. An *Isentropic* (constant entropy) process

b) **[Question]** Given that a water rocket expels the water in a very short period of time (a second or two) and that the gas must perform work on the water in this expulsion, which type of process above do you think most accurately models the expansion process of the gas?

c) **[Question]** Given an initial internal pressure, $p_0 = 7$ bar, an ambient atmospheric pressure, $p_a = 1$ bar, and the density of water $\rho_w = 1000$ kg/m³ what is the initial water exit velocity?

d) **[Question]** If the expulsion process were isobaric, derive the rocket ΔV as a function of ***water fill fraction***, $\alpha = \frac{V_w}{V_t}$ where V_w is the volume of the bottle initially filled with water and V_t is the total volume of the bottle. Weigh your bottle and altimeter using the scale in Prof. Cantwell's lab in order to compute this.

e) **[Question]** Since the expulsion process is not really isobaric, we would like to derive the ΔV for the isothermal and isentropic cases as well. Unfortunately, there is not a closed form solution for this so we will need to compute a differential equation for velocity increment and then integrate numerically. The Jupyter notebook linked here gives the background necessary to do this integration. *See the Appendix at the end for instructions on how to kick off a useable environment for this notebook that you can modify for your own computations.*

f) **[Question]** Given the results from this analysis, how would you pick the fill fraction α? How different is your answer for the isothermal and isentropic cases?

Problem 2 - Trajectory Simulation & Optimization

In the previous problem we looked at ΔV potential of your rocket. This is interesting and informative but it neglects to consider two unfortunate facts:

1. Your rocket must fight gravity
2. Your rocket will be slowed by aerodynamic drag

In this problem you will develop a very simple numerical simulation of the rocket's flight trajectory and use this to interrogate its performance sensitivity to some design parameters. In order to compute the 1D trajectory of the rocket numerically you must know three things:

1. It's mass at each point in time
2. The thrust it is producing at each point in time
3. The drag upon it at each point in time.

You have all the tools at your disposal to compute these things now with a few assumptions:

- Assume the drag coefficient (Cd) of the rocket is constant at 0.4. If you are brave, you can try and measure the drag coefficient for extra credit! Assume also the reference diameter for the drag calculations is 11cm (roughly the diameter of the 2L soda bottle).

- You will chose the nozzle diameter of your rocket eventually, but start by measuring the diameter of your nozzle in the lab.

In order to get you started, the Jupyter notebook I linked to above also solves a similar trajectory problem. Again, *see the Appendix at the end for instructions on how to kick off a useable environment for this notebook that you can modify for your own computations.*

Once up and running, there are two questions to answer:

a) **[Question]** Given the initial nozzle diameter, use the 1D trajectory simulation to pick the optimal α. Does it differ markedly from the simpler ΔV analysis?

b) **[Question]** Now look at optimizing over both nozzle diameter and α simultaneously. Should you modify the nozzle we've given you to achieve optimal performance?

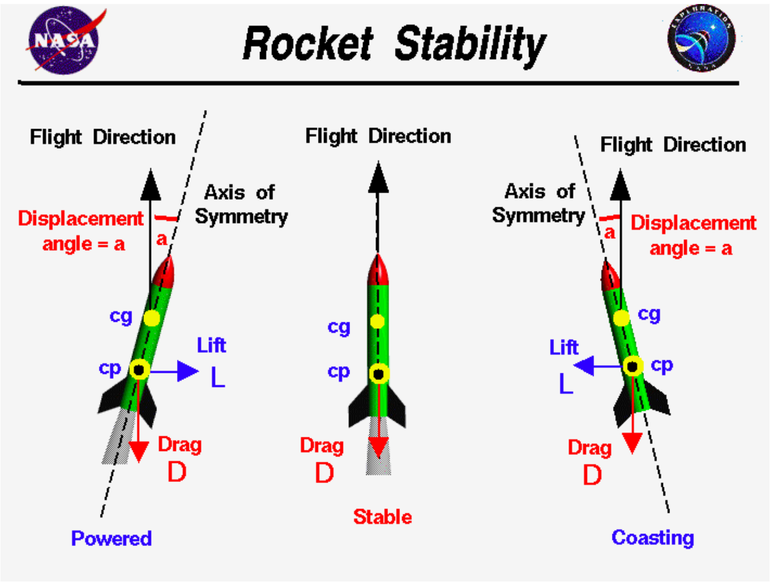
Problem 3 - Project description and CP/CG

The final project is pretty simple - you will be given a few parts for a nozzle and then tasked to:

1. Go buy one (or a couple) 2L soda's. Beverage choice is up to you.
2. Drink the soda (see the pictures for examples. Friends are recommended)

3. Build a water rocket from the empty bottle. I've included our journey as an example of something that produced a flying rocket but there are many ways to do this. The only strict requirement is that you must use the nozzle we've provided (although you can modify it's inner diameter through whatever means you determine necessary) and we will set the pressure in the rocket to 7 bar when we launch. I would recommend reading the rest of the description carefully.

A critical element of a successful water rocket are the fins. Fins stabilize the rocket's flight so that it flies straight. They do this by correcting there rockets *angle of attack* towards zero whenever it begins to point in a direction different than it's velocity direction:



In order to create the conditions in which a rocket self-stabilizes itself like this, we must design it such that the center of pressure (CP) is behind the Center of Gravity (CG) for the entire flight. Center of Gravity (CG) is the point at which the integrated force of gravity one side of the point exactly equals the integrated force on the other side. Analogously, center of pressure is the point along the rocket where the integral of the aerodynamic forces (pressure) on one side of the point exactly balance those on the other side. A body will rotate about its Center of Mass (CM) which is equivalent to the CG in a uniform gravity field. Looking at the picture above, when the CP is ahead of the CG, a small perturbation in the direction of flight of the rocket will cause a net aerodynamic torque reinforcing this movement, quickly causing the rocket to go unstable. When the CP is behind the CG, that same perturbation will generate a torque restoring the rocket direction back towards the velocity direction, stabilizing the rocket. So our goal is to design a rocket where the CP is always behind the CG.

Center of mass can be computed in closed form for our rocket assuming some simple geometry. The general definition for the location vector of the CM is:

$$\mathbf{x}_{cm} = \frac{1}{M} \iiint\limits_{Q} \rho(\mathbf{x})\mathbf{x} dV$$

Because we can reasonably assume the rocket is axisymmetric, we only need to compute this along the axial dimension of the rocket:

$$y_{cm} = \frac{1}{M} \int_0^l \rho_L(y) y dy$$

[**Question**] Assuming the water rocket is well represented by a hemispherical nozzle-end, a cylindrical main section, a set of fins with mass m_f centered at the base of the hemispherical end, a payload mass m_l and a bottle whose mass is uniformly distributed down its length with a linear density $\rho_l = \frac{m_b}{l}$, compute the axial location of the center of mass as a function of the height that the water is filled to in the bottle. Use the measured length and mass of your bottle and fins and $m_l = 15g$.

There are well documented approaches to computing CP as well but in practice it is easier to use a tool. OpenRocket is a nice open-sources software tool to estimate the center of pressure for a rocket you design. Note that you will need Java 8 Runtime to run OpenRocket (double check it's Version 8!).

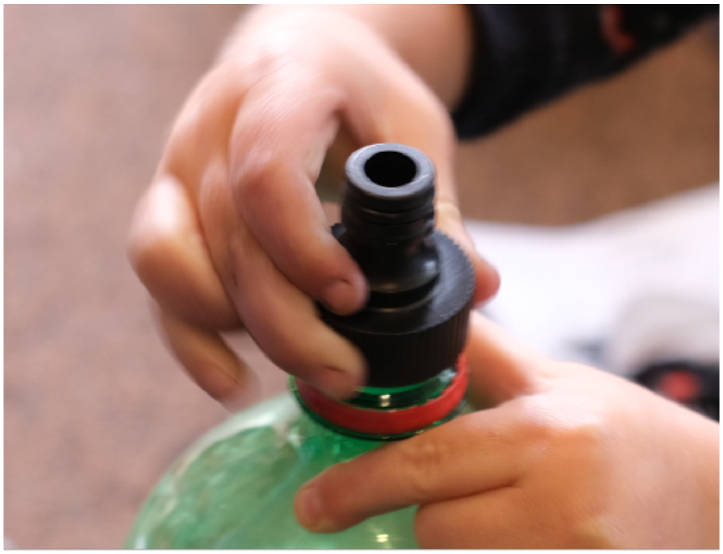
[**Question**] Using OpenRocket, model the water rocket and pick fin geometry that places the CP behind the CG of your rocket for all water fill levels.

Our build

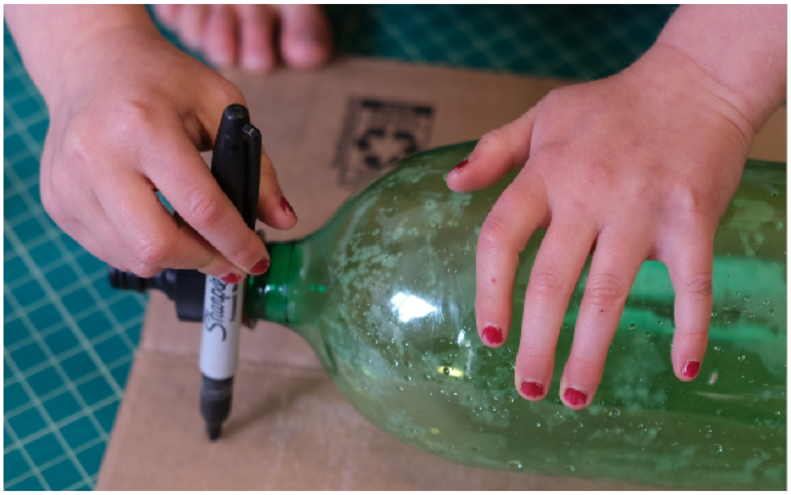
These are the materials we started with. Note the hot-glue gun is a "low-temp" type. This is important because if we damage the plastic bottle with heat it may blow up.

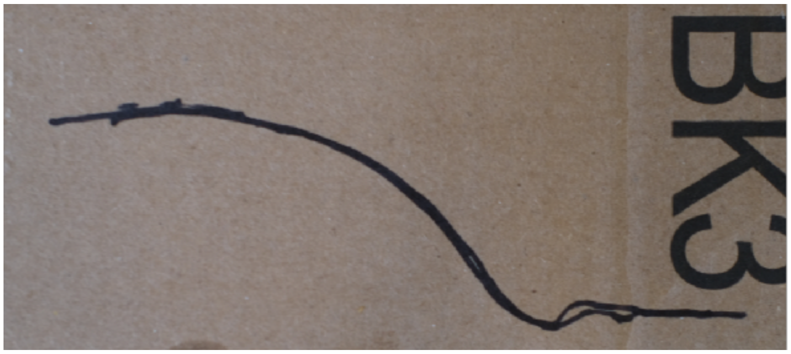
Although the nozzle comes with a rubber washer in it, we will need to add a second. You should have one in your kit. The reason is that the threads on the nozzle are not the same as the soda bottle threads so we can only get a couple of threads of engagement. We need the extra washer to seal the bottle when it doesn't bottom out well on just one.

Next screw on the nozzle firmly. Note that it will take a pretty good grip to really bottom it out on the washers. You can check pressure tightness by blowing in the nozzle and seeing if the assembly leaks at all.

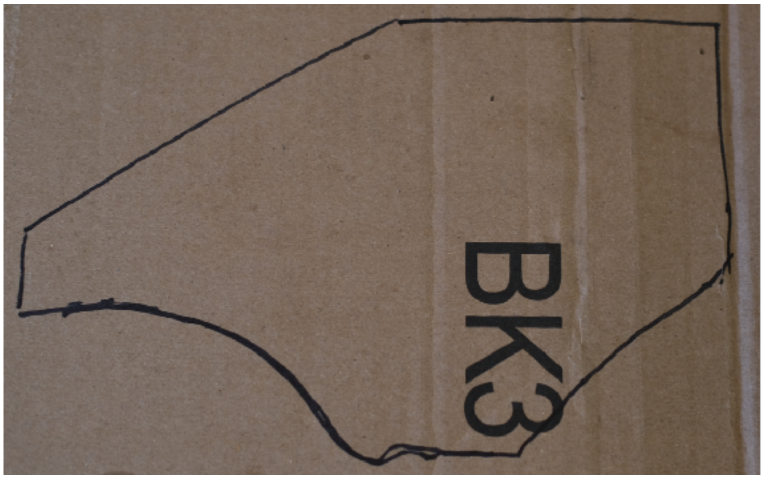


Next trace the contour of the bottle on some cardboard to start the fins. This is so that the fins conform to the contour of the bottle up to the nozzle where we will attach them.





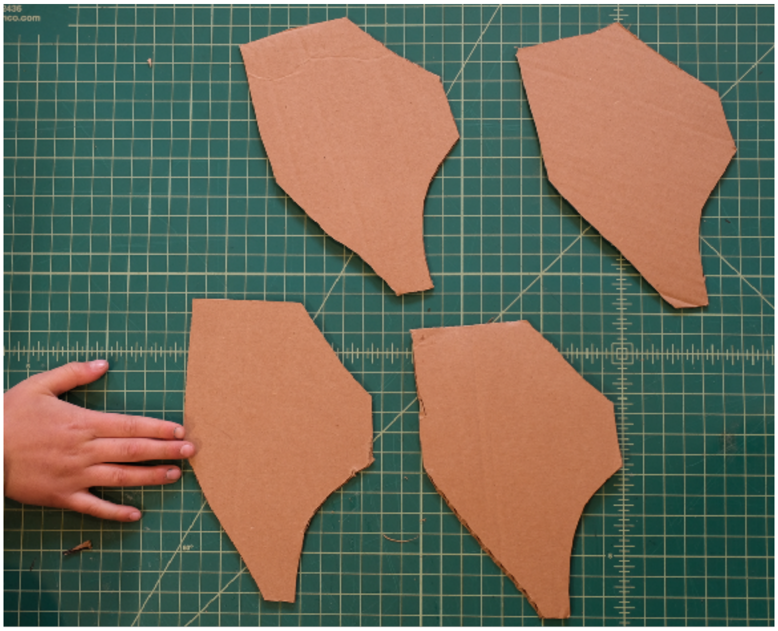
Now draw the rest of the fin however you see fit. **Be careful in picking your fin design!** The first design we flew straight for a while, then turned left and flew straight more and landed on a roof across the street. This is because the CG started ahead of the CP, moved behind it at some point (the rocket became unstable) and then moved back ahead again.



Note also that we are using cardboard for our fins. Other materials (such as foam core) are likely better but this is what we had. BEWARE THAT CARDBOARD IS NOT GOOD WHEN IT GETS WET. Given that this rocket will be shooting water everywhere, this is something to consider.

We cut out our four fins out with an exact knife. Cut one, trace 3 more so they are identical and then go.

Then do your best to hot-glue them to the bottle!



Lo Temp

Lo Temp

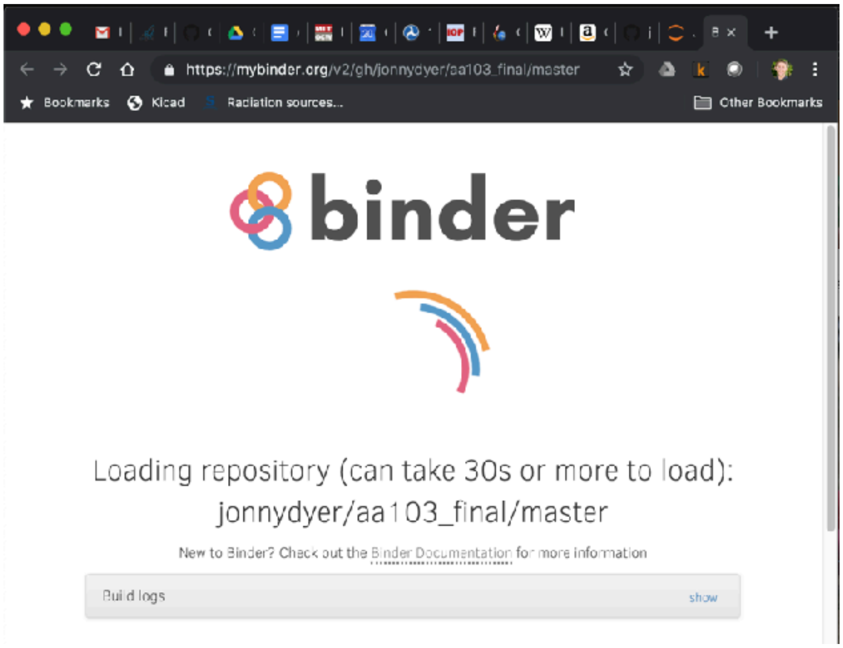
A final rocket and a preview of team operations at launch day!

Appendix - Running analysis in Jupyter

There are several ways to get up and running with Jupyter. For this project, the simplest is to use a service called Binder. I have checked in the example notebook at https://github.com/jonnydyer/aa103_final/blob/master/AA103FinalProjectExample.ipynb which means you can open, copy and modify it for your own purposes by going to:

https://mybinder.org/v2/gh/jonnydyer/aa103_final/master

When you click the link, the first thing you will see is a loading page:

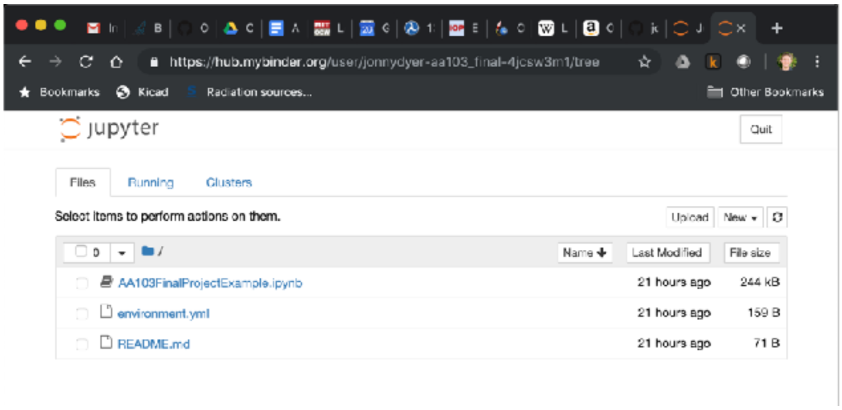


Loading repository (can take 30s or more to load):
jonnydyer/aa103_final/master

New to Binder? Check out the Binder Documentation for more information

Build logs show

Binder works by creating a dedicated, temporary virtual machine in the cloud with all of the necessary pre-requisites for Jupyter and then starts a Jupyter server for you. Once it has completed all of this you will see:



https://hub.mybinder.org/user/jonnydyer-aa103_final-4jcsw3m1/tree

Select items to perform actions on them.

Name	Last Modified	File size
AA103FinalProjectExample.ipynb	21 hours ago	244 kB
environment.yml	21 hours ago	159 B
README.md	21 hours ago	71 B

This is the Jupyter file browser showing you the files available to edit or interact with. Note that the URL in the browser has now changed (in this particular case to hub.mybinder.org/user/jonnydyer-aa103_final-4jcsw3m1/tree). **This URL is specifically for the virtual environment you just kicked off and is the one you should use to return to it if you close the tab!** This way you can make edits, save them and come back later to work more. The edits are only saved within the virtual machine you created when you originally clicked on the link above so if you lose this URL, you will also lose anything you added or changed from the starter file.

The first file in the list called *AA103FinalProjectEample.ipynb* is the one we want, so click on it and it will open in a new tab. You can now interact with the notebook. Double clicking on a text section lets you edit that text using the Markdown language. You can freely edit code cells and then run them by pushing the little play button at the top of the page or pressing Shift-Enter when the cursor is in the code cell.

That should get you up and running. I highly recommend downloading your notebook locally periodically as you work (you can do this from the File menu) just in case the service shuts your virtual environment down.

It really is nice to have a local python environment to do these problems. There are good instructions here to get up and running with Jupyter. The Anaconda graphical installer works well on Linux, Mac and Windows and gives you everything you need.

If you have questions, e-mail me at jonny.dyer@gmail.com and I'll find a time to have office hours over the next two weeks.