DISCLAIMER: There is code itself contains many comments and succinct explanations to the majority of the functions, constants, and even our thought process. It would be most beneficial to have the code open as you read through this.

Introduction:

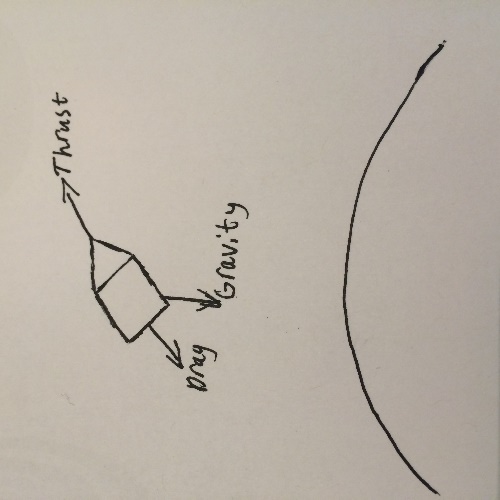
A typical phrase to describe a task’s difficulty is “Well, it’s not rocket science!” We decided to go and figure out for ourselves how difficult this ‘rocket science’ was, and if it was all it’s made out to be. Starting with some undefined mass, we were to determine how its motion would be effected if it were to be launched from the earth’s surface by some force. From this, a few forces and variables showed themselves as being necessary to launch a rocket.

First is the force of gravity, reliant on masses and height of the rocket. This is the force that keeps an orbiting rocket from flying off into who knows where. Next is the force of thrust, what the rocket is exerting on itself in order to propel into space. This is what gets us to an orbiting height and lets us reach the required velocity to maintain that orbit. Last is the force of drag, which slows down our approach to a desired height or velocity until we are clear of the atmosphere.

These sets of equations working in cohesion is what allows us to launch a rocket and predict its acceleration, velocity, and position for different times. Whether this result is a glorious orbit or falls into the ground half a mile away all depends on what we put into it. What we put into it is another set of variables that need to be determined. What fuel to use, at what rate to exhaust the fuel, when to cut it off. These are the questions that much of the time in this project was spent mulling over. Getting to the point when we can launch a rocket, have it exhaust its fuel and actively changing the forces because of the fuel loss; that is rocket science.

Description of Full Physical System

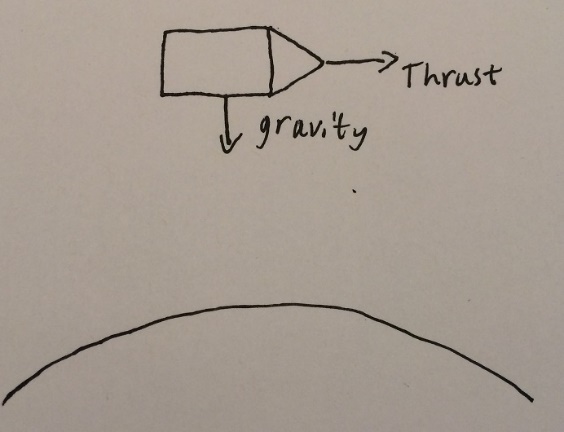
At Launch

 The rocket starts at the radius of the earth (6371 km) on the equator before launch. Before any thruster activate, the only force is gravity keeping it on the ground. Once the launch initiates, the rocket has three forces acting on it. These are the gravitational force, the thrust force, and the drag force. The gravitational force is pointing towards the center of the earth and is dependent on the mass of the rocket and its distance from the earth. Thrust is a function of the rate of fuel usage and gravitational acceleration, directed towards the way the rocket is pointing. Drag is a function of the air pressure, which changes with distance from earth’s surface, the innate drag value of the rocket design, and the current velocity of the rocket.

During Primary Thrusters

The sources of force on the rocket here don’t change, but the velocity and acceleration is increasing until the first thrusters run out. During this time, the rocket is tilting at a constant rate, allowing it to have a strong tangential velocity when the primary thrusters run out. This is a head start for when the secondary thrusters are firing, trying to reach the required speed for orbital velocity.

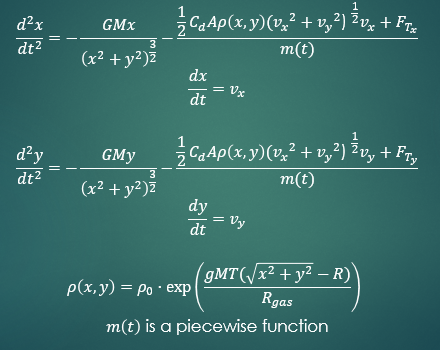
Secondary Thrusters

 Once the secondary thrusters are fired, the rocket has a force and acceleration going in the tangential direction, but at this point there is no longer any sources of drag as the rocket is too high up in the atmosphere. Only the secondary thruster force applied tangentially and the gravitational force inward are effecting the rocket. Secondary thrusters will keep running until the rocket reaches a required velocity to maintain orbit. At that point they will turn off.

Orbit

At this point the rocket has a high enough tangential speed to oppose the naturally force of gravity. The equation determining the necessary velocity needed to maintain orbit against gravity has been reached. Now, the rocket will proceed in its orbit without change or interruption unless some new force acts on it, throwing it out of balance and ending the circular orbit.

Below is the system of ODEs used in the simulation.



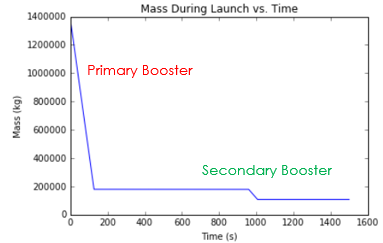
Code Description:

Things to note:

* We set the origin as the center of earth (approximated as a sphere).
* Parameters x and y are relative to the origin (**not** relative to the earth’s surface)
* i.e. (0, 0) is **not** recognized as a point on the earth’s surface.
* On the other hand, (0, 6371e3) **is** recognized as a surface point.
* Launches occur at the equator.
* Rocket orbits clockwise around earth
* The axes are positioned such that the earth's rotation is purely in the xy plane
* Earth's rotation is also clockwise

Function list:

* rk4\_step and rk4
  + These functions work together to integrate a given function using the fourth-order Runge-Kutta method. A full explanation and derivation of this algorithm is discussed in the textbook.
* force\_gravity
  + This function calculates the force the rocket experiences due to gravity. First, the force magnitude is calculated, and then it is split into its x and y components.
* air\_density
  + This function returns the air density at a certain radial height above earth’s surface. An equation modeling air density as a function of height pulled from Wikipedia is used here. The equation is slightly modified—shifted so that it works with our coordinate system
* force\_drag
  + This function calculated the force the rocket experiences due to drag. The x and y components are calculated explicitly
  + This function calls the air\_density function, since the drag force depends on air density.
* acceleration
  + This function sums up all of the forces and divides this net force by the mass of the rocket (which is changing in time). Since the forces are returned in x and y components, the acceleration is also returned in x and y components.
* mass
  + This function returns the mass of the rocket at a given time. The majority of the rocket's mass is due to fuel stored in its primary and secondary boosters. When these fire, the fuel is burned, and hence the rocket's mass will depleted over time. This is a piecewise function. **There are two functions that bear this title, and they serve different purposes.**
    - The first mass function to appear does not burn the secondary fuel. This function is used only to help determine when to start and stop the secondary boosters (See approximate\_boost\_info). Once the start/stop time is calculated, we can run the final simulation using this information, which will allow the rocket to reach orbit
    - The second mass function does burn the secondary fuel. This function is used in the final simulation that plots trajectory of the rocket during launch and orbit.

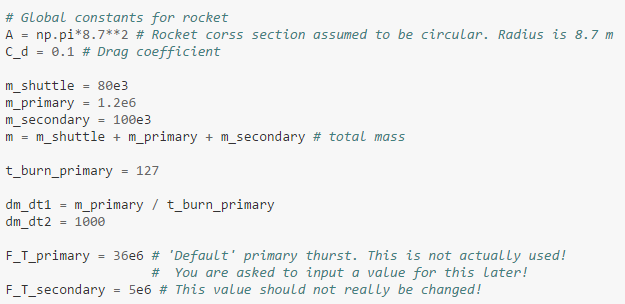


* force\_thrust
  + This function returns the thrust force of the rocket at a specific time. Like the mass function(s), it is piecewise. The primary thrusters are fired as to simulate the rocket tilting 90 degrees to the right at a constant rate. The secondary boosters are fired such that they are always directed tangent to earth. **Once again, there are two functions that bear this title, and they serve different purposes.**
    - The first thrust function to appear does not fire the secondary boosters. It is only used to help determine when to start and stop the secondary boosters (see approximate\_boost\_info).
    - The second thrust function does fire the secondary boosters. It is used in the final simulation that plots the trajectory of the rocket during launch and orbit.
* approximate\_boost\_info
  + The function runs the simulation without firing the secondary boosters in order to estimate when the secondary boosters should start and stop. The built-in odeint function is used to integrate during this simulation. A thorough explanation is provided in a markdown cell just before the function in the notebook.
* F
  + This function sets up our ode system for integration.
* print\_init
  + This function simply prints out all relevant constants and initial conditions of the launch.
* test
  + This function runs the final simulation and plots a variety of relations, including the trajectory (2D and 3D) and various velocities vs time. It calls all of the functions above and integrates using the fourth-order Runge-Kutta method.

Analytic Solution:

Our problem does not exactly have an analytic solution. More specifically, there really isn't a closed form solution that models the trajectory of the rocket, and this is primarily due to the effects of drag and variable air density at relatively low heights. Because of this, we have to find other ways to check the results.

Below is a picture of the rocket's relevant quantities. The values seen in the picture were not arbitrary; they are approximate values from actual missions to the International Space Station about 340 km about earth. With this in mind, there is one way to check if the system accurately models a rocket's trajectory...



Does the rocket with the provided specifications reach an orbit around 340 km off the earth's surface?

We do have analytic solutions for many of the test cases discussed later in the paper, but some do not have such solutions. In order to check our results in the latter case, plots were created analyzed accordingly.

Test Cases

Testing Gravity (1)

If an object is dropped from a relatively small height (i.e. 20 m), it’s acceleration should be essentially a constant 9.8 m/s^2. However, if that same object is dropped from a relatively large height (i.e. 200 km), the object’s acceleration should noticeably change as it falls.

One way to detect this change is to numerically calculate how long it takes the object to hit the ground after it is dropped from a given height. That is precisely what the first test case was.

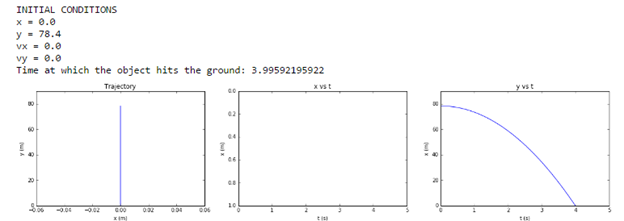
If gravity is constant, the object should hit the ground at



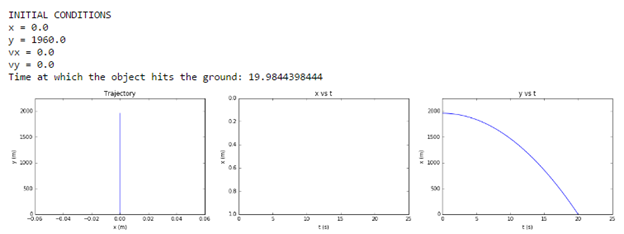
If it is not assumed to be constant, it should hit the ground later than t\_ground

Test Results:

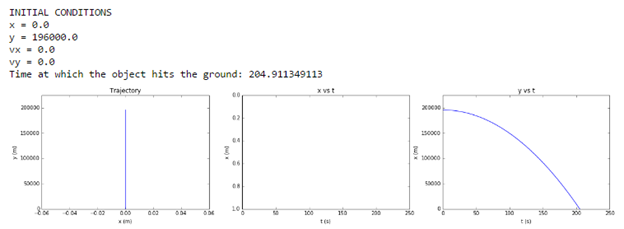
1) t\_ground = 4 s



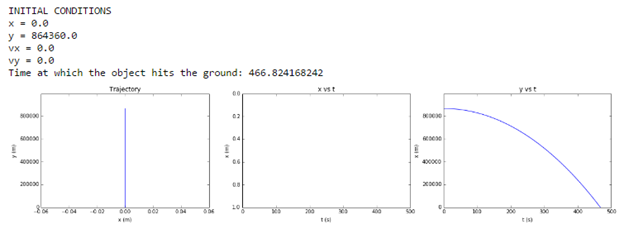
2) t\_ground = 20 s



3) t\_ground = 200 s



4) t\_ground = 420 s



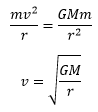
Gravity Test Results (1)

The first two tests were at relatively small heights, so the numeric value for the time at which the object hit the ground was approximately equal to t\_ground. However, the next two cases were at significantly larger heights. The time at which the object hit the ground was larger than t\_ground for these cases!

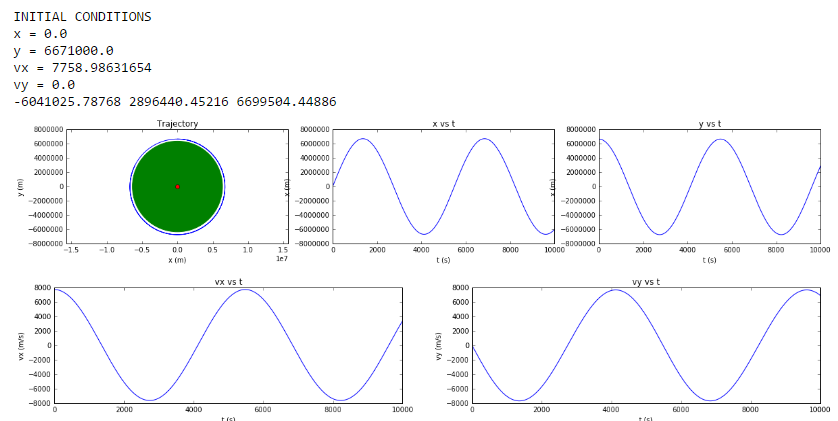
Gravity Tests (2)

One aditional test was also done. The idea was to see if we could get an object to orbit earth if we provided it with specific initial conditions to do so. This is essentially identical to a problem 3 on Homework 8.

The object was initial positioned a certain distance R from the center of earth purely in the y direction. An initial velocity provided purely in the x direction. To find the required initial velocity for the initial distance, set the force due to gravity equal to the centripetal force:

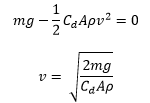


In the following test, the object was positioned r = 300 km above the earth's surface, so an initial x velocity of about 7759 m/s was provided.



Testing Drag with Constant/Variable Air Density

Here, gravity no longer changes and is kept at a constant 9.8 m/s^2. Considering drag forces with constant air density, a dropped object will reach a terminal velocity (assuming it has enough time to reach terminal velocity). This can be calculated easily as follows:

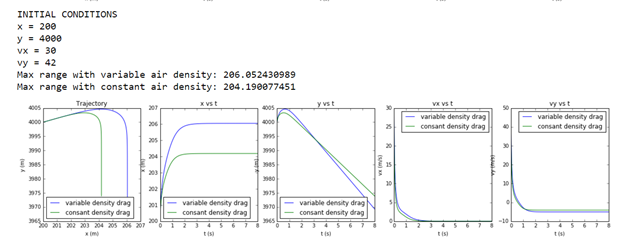


In the following tests, we set all constants (except g) equal to 1. Hence, the object’s terminal velocity should be sqrt(2\*9.8) = 4.43 m/s.

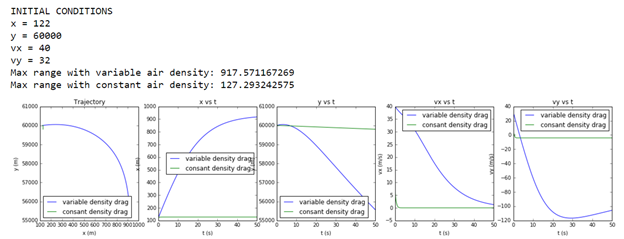
But let’s consider the effects of variable air density. Air density decreases exponentially in the troposphere, so higher altitudes will have lower densities. If there are fewer air particles holding back the dropped object, it can reach greater terminal velocities. Using this same reasoning, it’s easy to conclude that a thrown object will travel further and higher with a variable air density than with a constant air density.

In these next tests, terminal velocities and maximum ranges were numerically/graphically compared.

Test 1



Test 2



General Findings:

 Plots of x vs. t and y vs. t show that the object traveled further and higher with a variable air density. The printed numerical values of the maximum range also show this.

 Both vx vs. t and vy vs. t show the correct terminal velocity for constant air density (roughly 4.43 m/s).

 The terminal velocity in variable density is slightly faster than 4.43 m/s in the first test and significantly faster than 4.43 m/s in the next test.

 In the second test, the object velocity in the y direction starts *slowing down* at a certain point. This is when the object begins to enter regions of higher density and is therefore met with more resistance, slowing it down.

There are two more tests for this in the actual code. However, they do no show any new information and were excluded in this write up for the sake of simplicity. Overall, the findings are completely consistent with what was expected.

Developing a Thrust Function

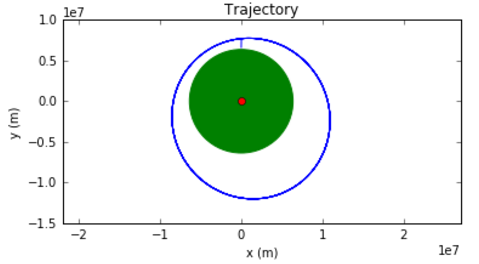
This proved to be the most difficult part of the simulation. The rocket needed to tilt in such a way that would give it a substantial radial velocity to get it to orbital height and just the right amount of tangential velocity to maintain an orbit. It sounds relatively simple, but once we actually tried to code this, we quickly realized this was a bit more complex than we initial thought.

We did not really have test cases to develop this function. It was a long road of trying out a bunch of possible thrust mechanisms to get the rocket into orbit and you'll find several notebooks of code focused on its development. It's important to stress here that this development was not a planned out process; we simply used trial and error and hoped to develop some intuition. Below is out general thought process.

Mechanism 1

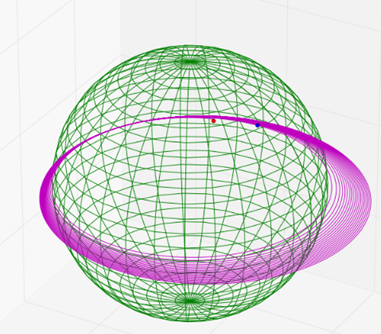
Launch the rocket vertically and instantaneously tilt it 90 once a sufficient height is reach. Then, provide an instantaneous thrust force that is strong enough to get the rocket into orbit (*any* orbit).

This was the beginning of the journey. We knew that this mechanism was horribly unrealistic, but we just wanted it to work.



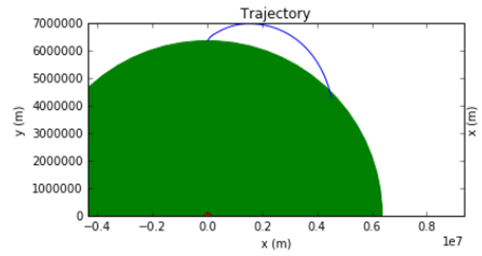
Mechanism 2

This is almost idential to Mechanism 1, except now the second thurst force is gradual rather than instantaneous. In the following example, we didn't get enough tangential velocity, so we spiraled towards earth.



Mechanism 3

Position the rocket vertically tilt the rocket 90 degrees at a constant rate. This was good, except we did get enough tangential velocity, so orbit was not possible.

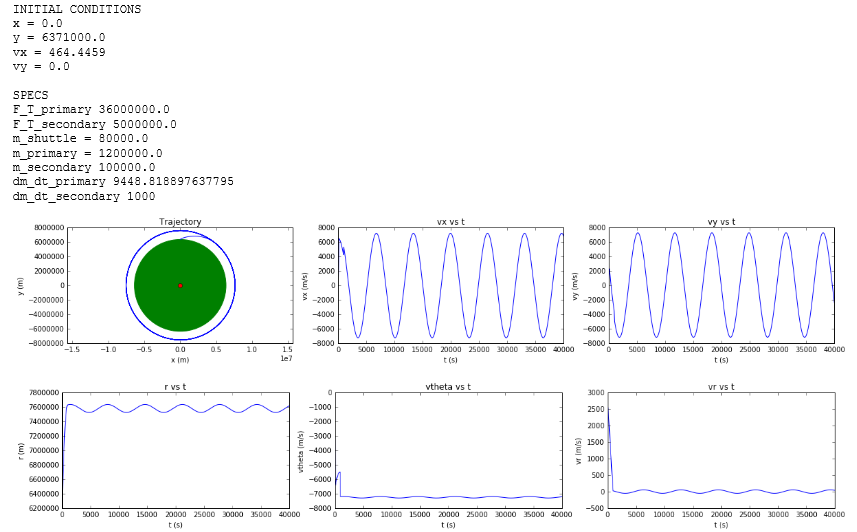


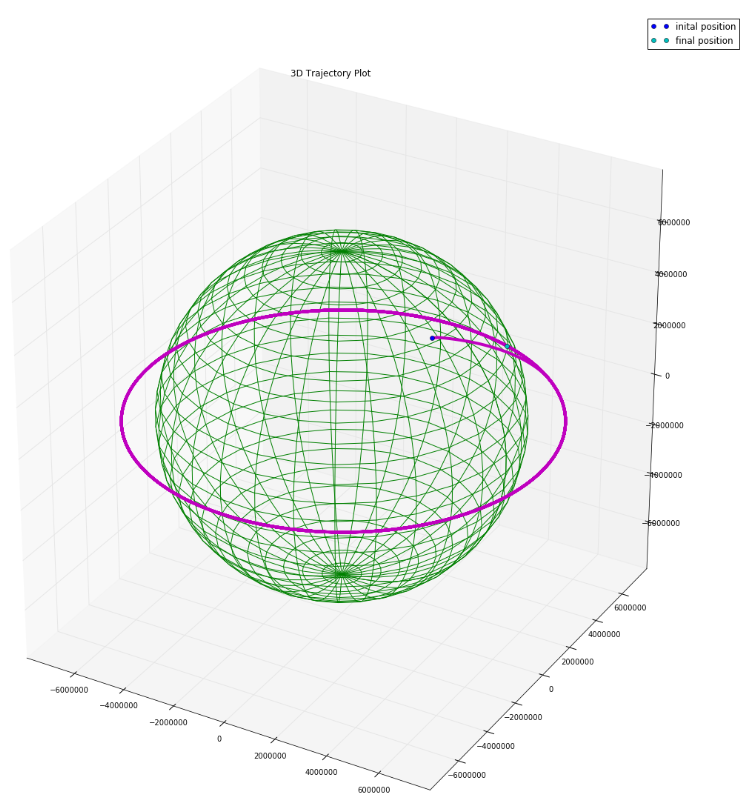
The final mechanism and the thought process is described in the code.

Final Results

These are the results after running the full simulation with the specs mentioned previous (and listed below for convenience). The rocket enters orbit at an orbital height of about 1300 km, which is still technically a low earth orbit (160 km – 2000 km), but this is significantly larger than the 'analytic' 340 km. There is much to discuss regarding this result, and this discussion will focus primarily on the devised thrust mechanism and a number of simplifications.

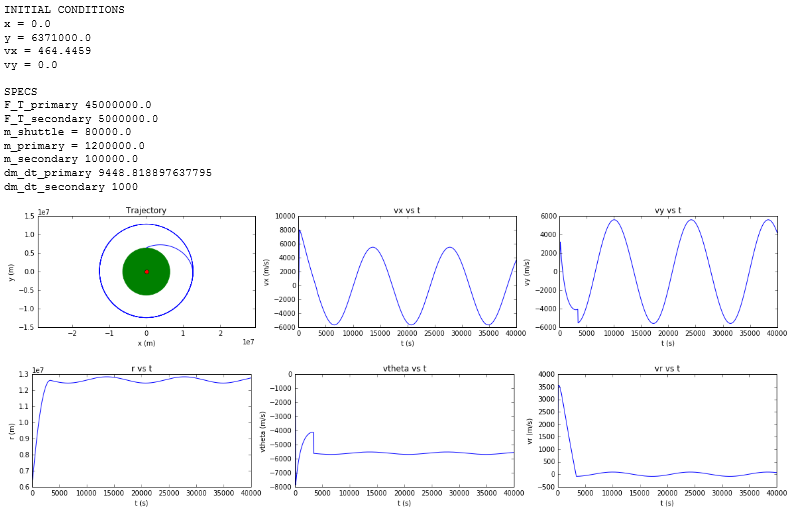
Simulation 1

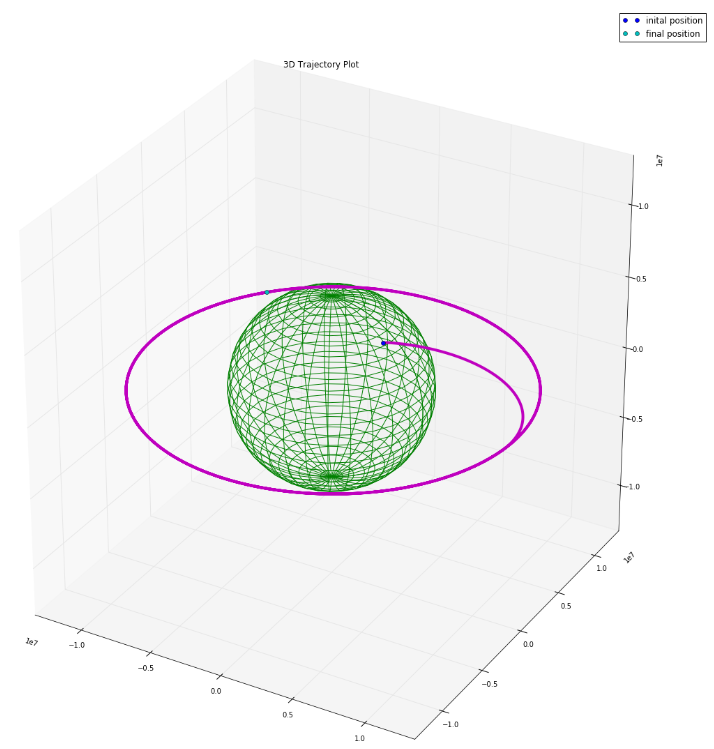




What happens if we increase the primary thrust force? The rocket should be able to get into a higher orbit. The following simulation displays the results after increasing the primary thrust force by 25%, and the they agree with our expectation. The rocket's orbital height increased by roughly 500 km.

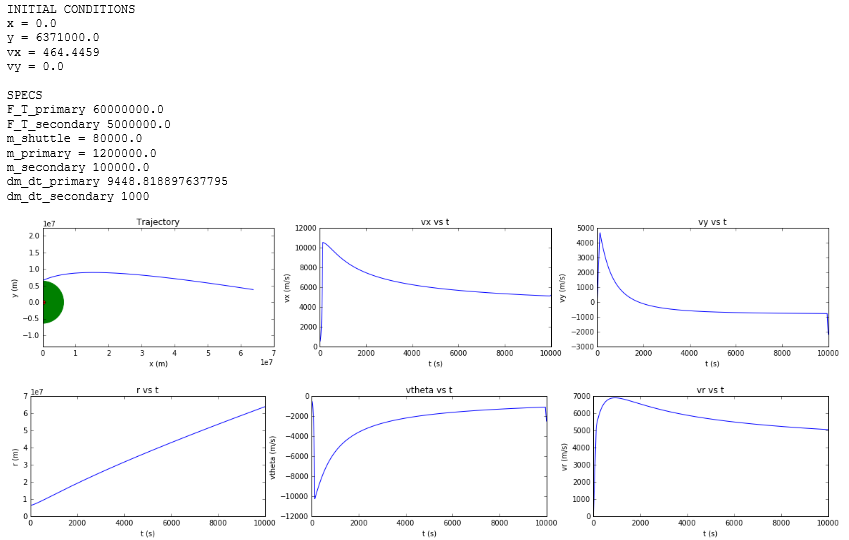
Simulation 2

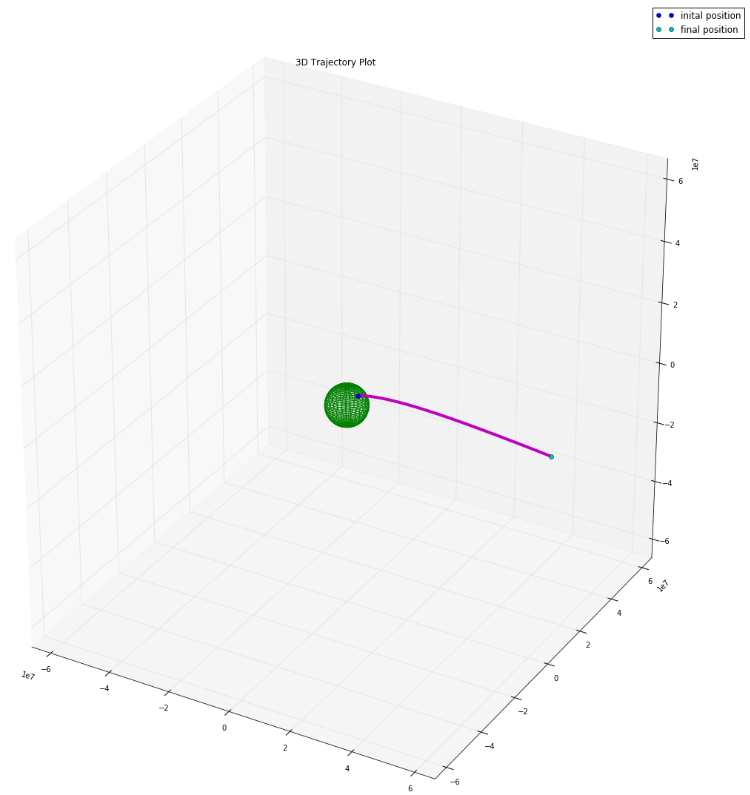




If we increase the primary thrust force enough, the rocket should be able to "escape" the gravitational pull of earth! A massive 60e6 N of thrust seems to do just that. It's very clear that the rocket is **not** coming back to earth.

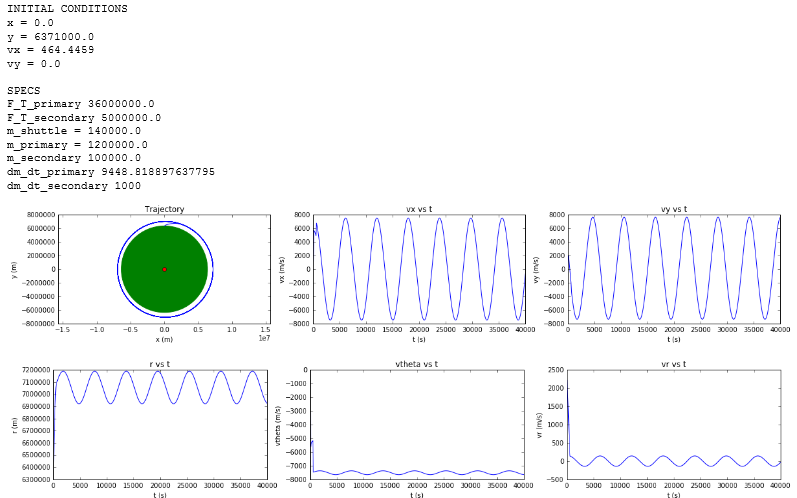
Simulation 3





Now suppose we return the primary thrust force to it's original (Simulation 1) 36e6 N and increase the mass of the shuttle (not the entire rocket!) by 75%. We expect the orbital height to be smaller in this case. The following simulation puts this to the test, and it looks like our expectations are correct. The orbital height is roughly 700 km off the earth's surface, or 600 km lower than Simulation 1's orbit.

Simulation 4



Discussion

Generally speaking, the results were solid; the we were able to get the rocket into orbit, and we were able to do so for a wide range of primary thrust forces. However our orbital heights were *significantly* higher than expected. When we simulated a launch with specifications similar to that used for a mission to the International space station, we got about 1000 km higher than the height of the ISS!

What's causing that? The next section offers much insight into that question. In brief, it is due to a simplification in the forces acting on the rocket. In reality, the forces are significantly stronger.

The next logical question to ask is why does the orbital radius vary sinusoidally in time? The first explanation that came to mind was that this might be due to approximations made during integration. However, this proved to be incorrect. We did a rerun of simulation 1 but with three times as many points and the results did not change at all. So, there must be something else causing these relatively small oscillations.

As it turns out, the secondary thrusters are the source of these oscillations. Now would be a good time to look at the markdown cell in the code explaining the approximate\_boost\_info function.

The start and stop time for the secondary boosters were calculated based off of the maximum radial height achieved without them. What we did not account for was the fact that firing the secondary boosters cause a tangential acceleration which in turn causes the radius to increase as well.

In brief, the maximum radial height achieve with the secondary boosters is greater than that achieved without them. So, the rocket is traveling at a speed required for a circular orbit at a height *slightly smaller than it's actual height*. Hence, the orbit is slightly elliptical.

This isn't a huge problem, because *all* real orbits are at least slightly elliptical.

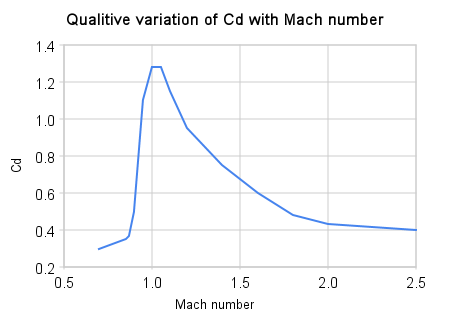
This also explains why the radial velocity oscillates around 0 and why the tangential velocity oscillate around ~7500 m/s.

Future Directions

With this project we were able to achieve a stable orbit from being given a force for the primary thrusters. All other variables are determined from that number, reaching any number of different orbit heights. What would make the program more usable in a situation would be to let an orbital height be inputted by the user. The program could be made to calculate the minimum of fuel needed in the primary and secondary thrusters to reach the stated height and required velocity for orbit. This would involve many more lines of code along with changing up how the entire process works, but it does seem doable. This would be a much more logical thing to do—input our *goal* and *calculate* the required specifications rather than inputting the specifications and calculating the results.

There were other variables that we more or less ignored in order to pursue the goal of actually getting the rocket into orbit, either because it was too complicated for us to code or we didn’t know how it worked. One was the presence of atmospheric density, which would apply additional drag on to the rocket. We only accounted for the air density within the troposphere, not the layers of atmosphere we must pass through when exiting the atmosphere.

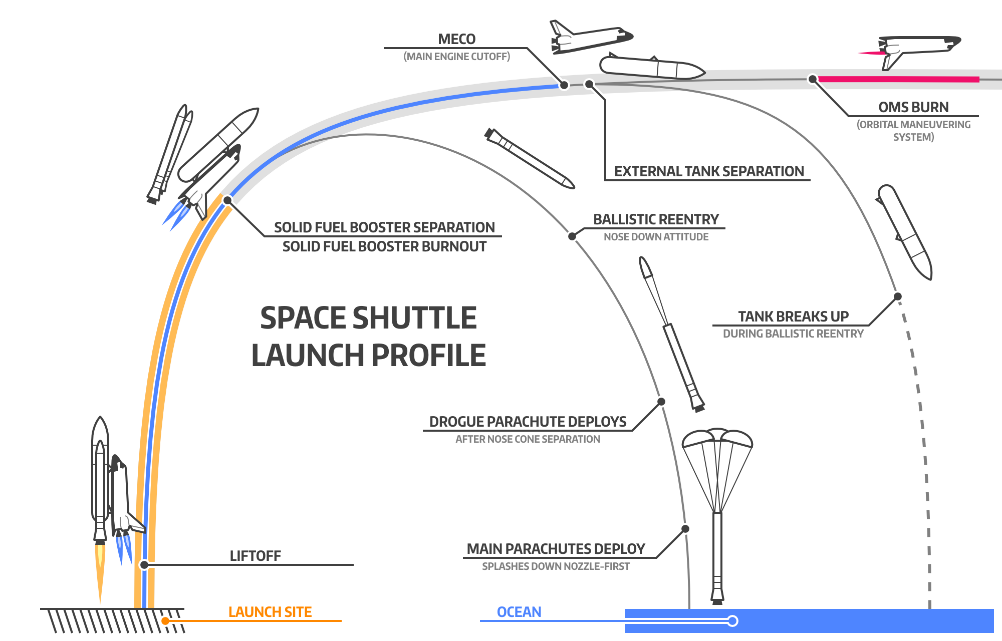
Another was our failure to account for how the drag changed once the rocket had hit supersonic speeds, as the amount of drag no longer behaves regularly and is instead stronger by a large margin. The drag coefficient increases and peaks and Mach 1, then decreases and levels out at .4. This is still stronger than the drag coefficient we were using the entire time. Doing this would involve us choosing which drag coefficient was being used, based on the velocity, along with creating a function for its calculation.

https://en.wikipedia.org/wiki/Drag\_(physics)#/media/File:Qualitive\_variation\_of\_cd\_with\_mach\_number.png

The general launch process was unrealistic. When our primary thrusters ran out of fuel, we made it as if its container "burned" too. In reality, just the fuel would burn, and the containers would be jettisoned afterwards. So, the decreasing mass was not accurately implemented. Furthermore, fuel was not continuously burned. There was a large gap between when the primary thrusters stopped and the secondary thrusters started. This is simply unrealistic. In real launches, boosters are continuously fired and they do not stop firing until the rocket is actually in orbit.

On the topic of extra variables we didn’t account for is that we treated the earth as a uniform mass for the sake of gravity calculations. This is not how the earth is set up, since its's density changes throughout.

Another thing that could have been done better were we not generalizing so many of the variables, is the tilting of the rocket. For our system the rocket begins its slow tilt the moment it launches, becoming horizontal once the primary thrusters have fully run out of fuel. In reality it is a much more hyperbolic function, barely tilting at all during the first stage of the rocket before the solid fuel booster is released from the system. The variance between our systems implemented is great, mostly due to the fact that our rocket, although simple, is horribly inefficient. This more complicated and realistic process is something that would have to be implemented if fuel conservation were ever to be a concern.



There was one glaring error with our entire set up though, that we should deal with. Unless the rockets purpose was to send up a satellite, we just abandoned who knows how many people to their deaths. These poor unfortunate souls, destined to float around earth for the next millennia, unless we were nice enough to give them enough fuel to break orbit and get back. For this we would have to calculate how much force over a time is needed to break stability and start the rockets decent towards the surface. Maybe then we wouldn’t be sending anyone on a one way ticket to eternity in orbit.

References

* <https://en.wikipedia.org/wiki/Rocket_engine>
* <http://sciencelearn.org.nz/Contexts/Rockets/Looking-Closer/Calculating-rocket-acceleration>
* <https://en.wikipedia.org/wiki/Density_of_air>
* <https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-07-dynamics-fall-2009/lecture-notes/MIT16_07F09_Lec05.pdf>