

Homework 3: We become rocket scientists

ENM1050, UPenn

Due date: September 30th by midnight (11:59pm)

This is an **individual assignment**. Submit your answers on Canvas using the instructions at the end of the handout. Late submissions will be accepted until midnight of the following Wednesday (11:59pm), but will be penalized by 10% for each partial or full date late. After the late deadline, no further assignments may be submitted; post a private message on Ed to request an extension if you need one due to a special situation such as illness.

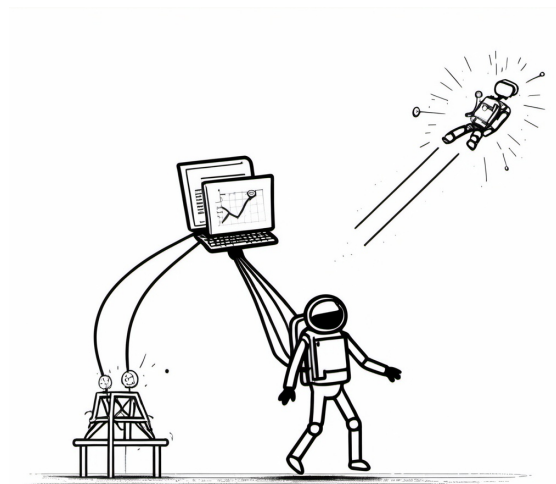


Figure 1: Dall-E thinks this looks like a "stick figure with a jetpack being launched by a catapult under projectile motion".

You may talk with other students about this assignment, ask the teaching team questions, use a calculator and other tools, and consult outside sources such as the Internet. When you get stuck, post a question on Ed or go to office hours!

Ballistics and rocket packs.

For a projectile undergoing projectile motion, an object of mass m and initial position $\langle x_0, y_0 \rangle$ is launched into the air with initial velocity $\mathbf{v} = \langle u, v \rangle$, and its trajectory evolves only under interaction with gravity. For this setting, the equations of motion are given by

$$\dot{x} = u \quad (1)$$

$$\dot{y} = v \quad (2)$$

$$m\dot{u} = 0 \quad (3)$$

$$m\dot{v} = -mg, \quad (4)$$

where $g = 9.8m/s^2$. For this case, we can check that this *initial value problem* is satisfied by the following trajectory.

$$x(t) = x_0 + u_0 t \quad (5)$$

$$y(t) = y_0 + v_0 t - \frac{1}{2}gt^2 \quad (6)$$

To make things a little more complicated, we will assume our projectile is a little astronaut with a jetpack containing initial fuel of mass m_f . We can activate our jetpack at a time t_{jp} - if we do this we will burn fuel at a rate α in exchange for generating thrust $\mathbf{F}_{jp} = \langle F \cos \theta, F \sin \theta \rangle$, where F is a given magnitude and $\theta \in [0, 2\pi]$ is a direction we are free to pick to point the jetpack. In this case, our initial value problem becomes more complicated. For $t > t_{jp}$, the governing equations change to

$$\dot{x} = u \quad (7)$$

$$\dot{y} = v \quad (8)$$

$$\dot{m}_f = -\alpha \quad (9)$$

$$\dot{u} = 0 + \frac{F \cos \theta}{m + m_f} \quad (10)$$

$$\dot{v} = -g + \frac{F \sin \theta}{m + m_f}, \quad (11)$$

for as long as $m_f > 0$, before reverting back to the initial equations. When the person hits the ground ($y \leq 0$, for $t > 0$) the trajectory finishes. The equations of projectile motion don't apply here unfortunately, so we will need to solve them numerically. To keep things simple in this assignment, we will arbitrarily set $\alpha = 1$, $m = 1$ and $F = 2$, which is completely unphysical. When choosing initial conditions, we will set $x(t = 0) = y(t = 0) = 0$, $u(t = 0) = v(t = 0) = \sqrt{2}$ and $m_f(t = 0) = 1$.

Your assignment.

Your task this week is to build up a simulator for a projectile which solves the governing equations and use it to launch our little friend as far as possible, picking θ and t_{jp} to maximize his distance traveled before hitting the ground. Mathematically, we would write that as

$$\max_{t_{jp} > 0, \theta \in [0, 2\pi]} x(t; t_{jp}, \theta) \quad (12)$$

Here, we use a semicolon to distinguish between the dependent variables in the solution (in this case the time t) and the parameters to be optimized over. Underneath the min we specify the range of values that we allow the parameters to take.

Action items.

1. Start a new Colab notebook. Add a title and your name to the top of the notebook, and acknowledge resources used. Refer to class notebooks for examples about how to format your notebook so that it effectively communicates your work.
2. Use calculus to confirm that Equations 5 are a solution of the initial value problem (Equations 7).
 - The derivatives balance the terms on the right hand side of Equations 7.
 - The solution at $t = 0$ matches the initial conditions of our system.

You can include this in your submitted notebook by taking a picture of your derivation with your phone and copy/pasting into a text block in the notebook.

3. Implement the explicit Euler method to solve the equations of projectile motion for the given initial conditions. (i.e. no rocket packs yet). Use a function to encapsulate the right hand side of the differential equation, so that it takes the solution and time at a given timestep, along with any parameters, and returns the derivatives of the solution at that time.
4. Use the analytic solution to verify that your code is correct, and that for sufficiently small numbers of timesteps you can reproduce the true solution. You can do this by generating a plot showing the true x,y trajectory's evolution over time as a dashed lined, and then super imposing numerical solutions for 8,16,32,... timesteps until they match.
5. Now that you have confirmed you have a working solver, add conditional statements and other modifications as needed to introduce the jetpack equations.
6. Use corner cases and limiting behavior to build your trust in the code that you've written. For example, run with $\alpha, F, m_f \rightarrow 0$ to confirm you recover the analytic trajectory from before.
7. Now that you trust your code, do experiments to identify optimal values for t_{jp} and θ . Make sure you have a text box labeled "Optimization approach" which clearly articulates how you achieved your optimal value, provides the optimal values as well as the corresponding x coordinate when $y = 0$, and a single code block which will reproduce this result. **The highest scoring value in the class will receive an extra point on your final grade!**
8. Did you remember to acknowledge your collaborators?
9. Submit your work as both an ipynb and pdf to Canvas.

Note: If you use an off-the-shelf integrator like `odeint` from SciPy you will face issues. You are welcome to attempt this, but please be careful to note that many libraries assume the RHS is continuous, and will break if you attempt to apply the discontinuous RHS coming from the jetpack problem. Make sure to confirm using either test cases or other techniques that any libraries you use produce a valid prediction, or you will not receive credit.