Modeling the Gravitational Slingshot Effect

Sam Cochran, Jacob Murri, and Caleb Wilson

December 2020

While we love the pdf format, unfortunately we cannot display animations. You can find our code and relevant animations on the following GitHub repository:

`https://samcochran.github.io/Gravitational-Slingshot/`

1 Introduction

In our project we are attempting to model the physics of the slingshot effect, whereby a satellite can swing around a planet and receive a boost in speed so that it can successfully reach its destination. In building a model for the slingshot effect, we start out with a lot to work with, as Newton derived the fundamental physics describing the laws of gravity over 300 years ago. All of our models rely on the predictions of Newton's Laws, but throughout the course of this paper we refine our assumptions to see how they affect the predictions of the model.

As a first step, we consider the general case of Newton's laws applied to n bodies which interact gravitationally. Let $\mathbf{x}_1(t), \mathbf{x}_2(t), \ldots, \mathbf{x}_n(t) : [0, \infty) \to \mathbb{R}^3$ be the positions of these n bodies, which have masses $m_1, m_2, \ldots, m_n \geq 0$. Then if G is the universal gravitational constant, the equations of motion given by Newton's Second Law are

$$\mathbf{x}_i''(t) = \sum_{j=1, j\neq i}^{n} \frac{Gm_j(\mathbf{x}_j(t) - \mathbf{x}_i(t))}{\|\mathbf{x}_j(t) - \mathbf{x}_i(t)\|^3}, \qquad 1 \leq i \leq n$$

In order to make our system easier to analyze numerically, we can introduce dimensionless time, space, and mass parameters to make our system more tractable. Let L, Θ, and M be characteristic length, time, and mass scales, with units of meters, seconds, and kilograms, respectively. Then introduce the dimensionless parameters $\mathbf{u}_i = \mathbf{x}_i/L$, $\tau = t/\Theta$, and $\mu_i = m_i/M$. Then we can rewrite the equations as

$$\frac{L}{\Theta^2}\mathbf{u}_i''(\tau) = \sum_{j=1, j\neq i}^{n} \frac{GM\mu_j L(\mathbf{u}_j(\tau) - \mathbf{u}_i(\tau))}{L^3\|\mathbf{u}_j(\tau) - \mathbf{u}_i(\tau)\|^3}, \qquad 1 \leq i \leq n,$$

which we can simplify to

$$\mathbf{u}_i''(\tau) = \frac{GM\Theta^2}{L^3} \sum_{j=1, j\neq i}^{n} \frac{\mu_j(\mathbf{u}_j(\tau) - \mathbf{u}_i(\tau))}{\|\mathbf{u}_j(\tau) - \mathbf{u}_i(\tau)\|^3}, \qquad 1 \leq i \leq n.$$

Then by choosing our mass, length, and time scales appropriately, we can set $GM\Theta^2/L^3 = 1$, so that our equations have the dimensionless form

$$\mathbf{u}_i''(\tau) = \sum_{j=1, j\neq i}^{n} \frac{\mu_j(\mathbf{u}_j(\tau) - \mathbf{u}_i(\tau))}{\|\mathbf{u}_j(\tau) - \mathbf{u}_i(\tau)\|^3}, \qquad 1 \leq i \leq n.$$

To make the equations easier to integrate numerically, we can make our system first order by setting $\mathbf{v}_i(\tau) = \mathbf{u}_i'(\tau)$. Then we have the equations

$$\mathbf{u}_i'(\tau) = \mathbf{v}_i(\tau); \qquad \mathbf{v}_i'(\tau) = \sum_{j=1, j\neq i}^{n} \frac{\mu_j(\mathbf{u}_j(\tau) - \mathbf{u}_i(\tau))}{\|\mathbf{u}_j(\tau) - \mathbf{u}_i(\tau)\|^3}, \qquad\qquad 1 \leq i \leq n.$$

2 Initial Models

In a real gravitational slingshot performed, for example, with a satellite, the mass of the satellite is far smaller than the mass of of the body it is being slingshotted around. For instance, the mass of any satellite is a very insignificant fraction of the mass of Jupiter. Hence, it is a good approximation to say that the motion of the satellite does not affect the motion of Jupiter or the Sun. For this reason, we choose an initial model where the mass of the satellite is zero (so that it is affected by the other relevant bodies, but it does not affect them).

For our initial model, we choose a setup where $m_1 = 1, m_2 = 1, m_3 = 0$. We choose the case of equal masses for the first two bodies because it is a very simple case.

Given the system of ODEs above (in the Introduction section), the masses listed above, and initial conditions (comprising initial positions and initial velocities), we can numerically integrate to determine the trajectories over time. This code that performs the numerical integration is found on our GitHub (link at top of paper) in the file `Code/simulation.py`. Now we will use the code we have written to create some figures which illustrate the dynamics of our system.

In each of these cases, there is something wrong with the outcome. The plots below, more than anything, illustrate the difficulty of choosing initial conditions that allow for a successful slingshot maneuver. Our task moving forward is to find a way to choose good initial conditions that allow for successful slingshot maneuvers.

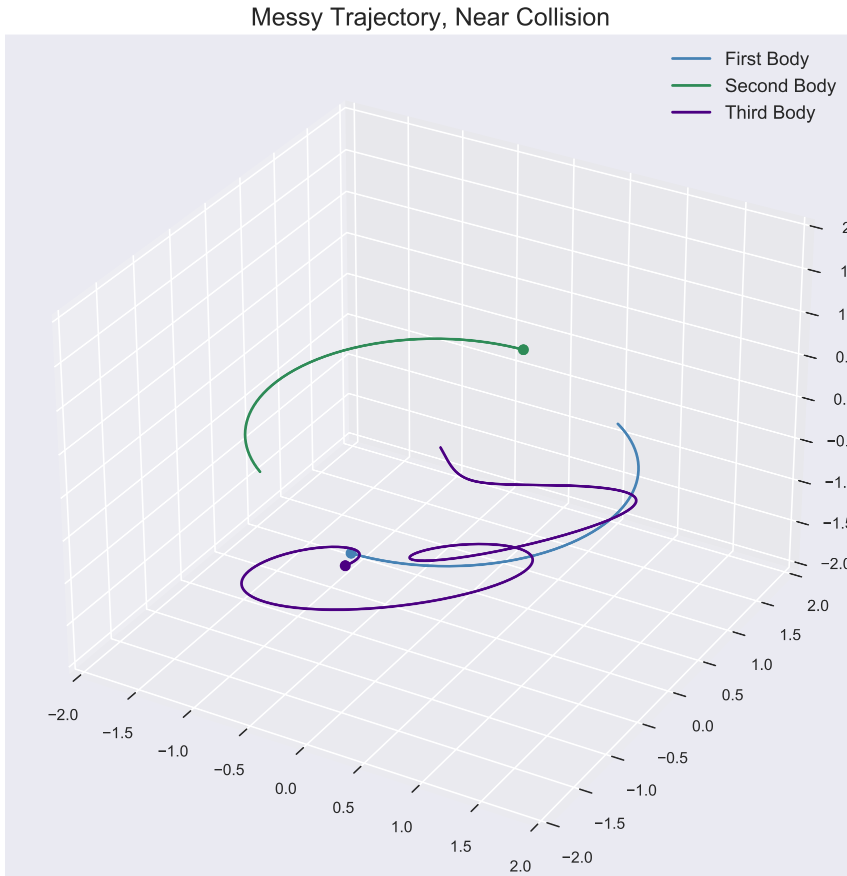

Figure 1: This model shows a messy trajectory achieved by choosing specific initial conditions for the toy problem with equal masses. It is interesting to watch how the third body moves in tandem with one of the larger primaries, oscillating aroudn it as both orbit the barycenter.

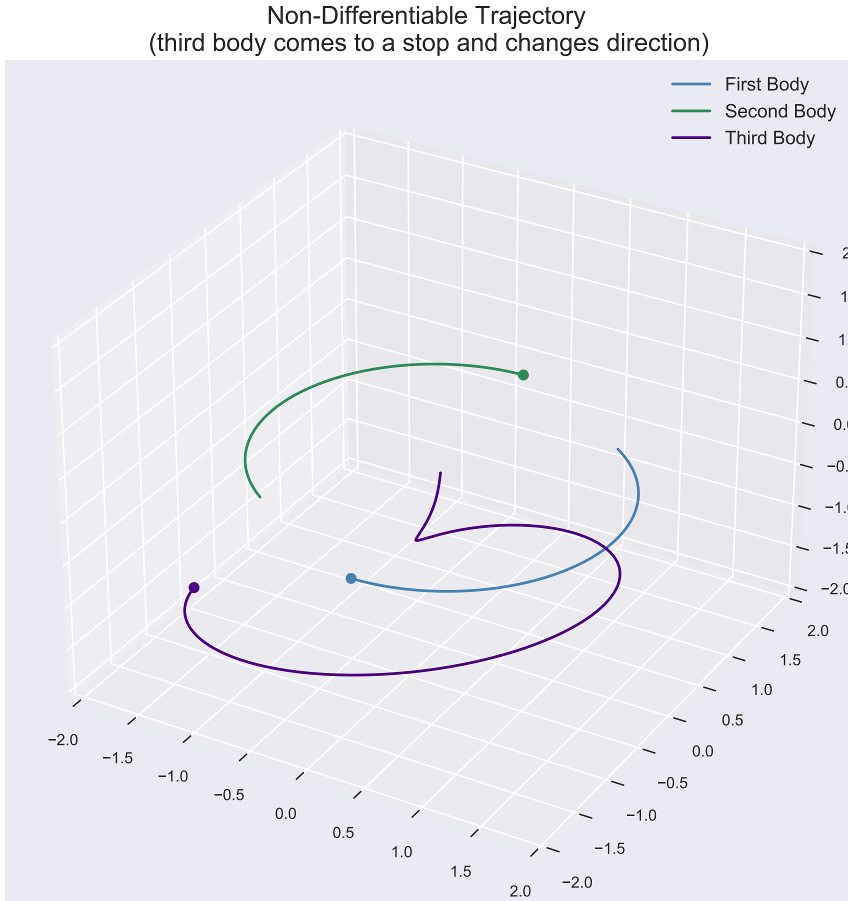

Figure 2: Interestingly, these initial conditions cause the third body to begin to move in one direction, then come to a complete stop before moving back to follow one of the primaries. The result is a jagged trajectory.

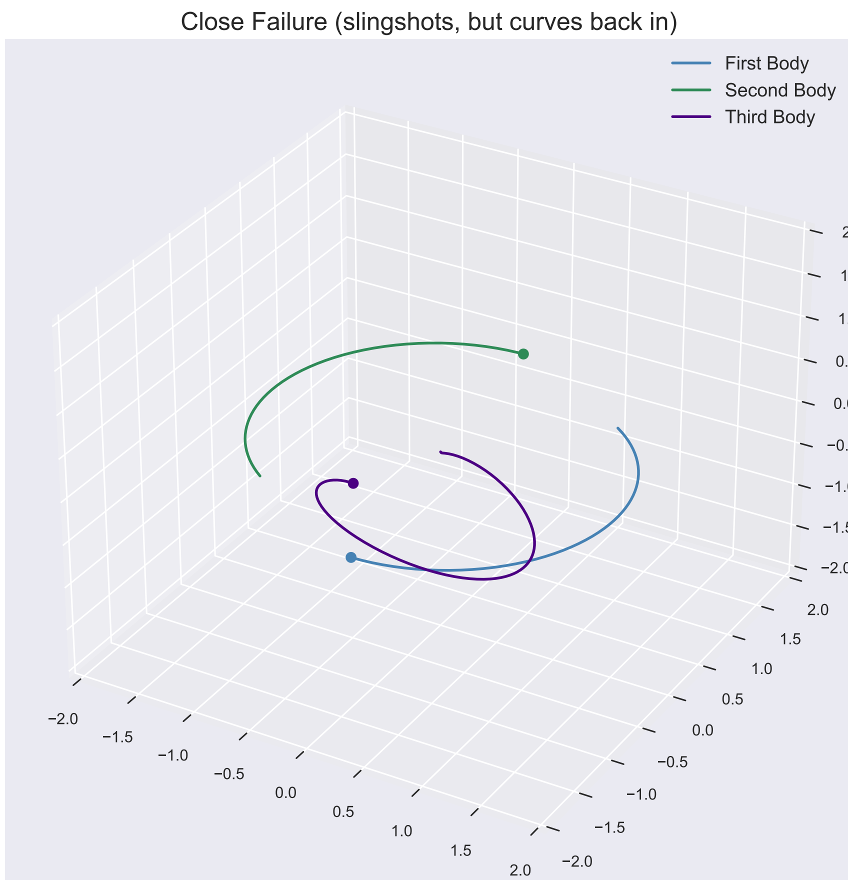

Figure 3: Here, we finally found a set of initial conditions that came close to allowing for a successful slingshot, though the third body curves back in. Still, it does gain a lot of velocity through the maneuver, which is what we are trying to capture in our model.

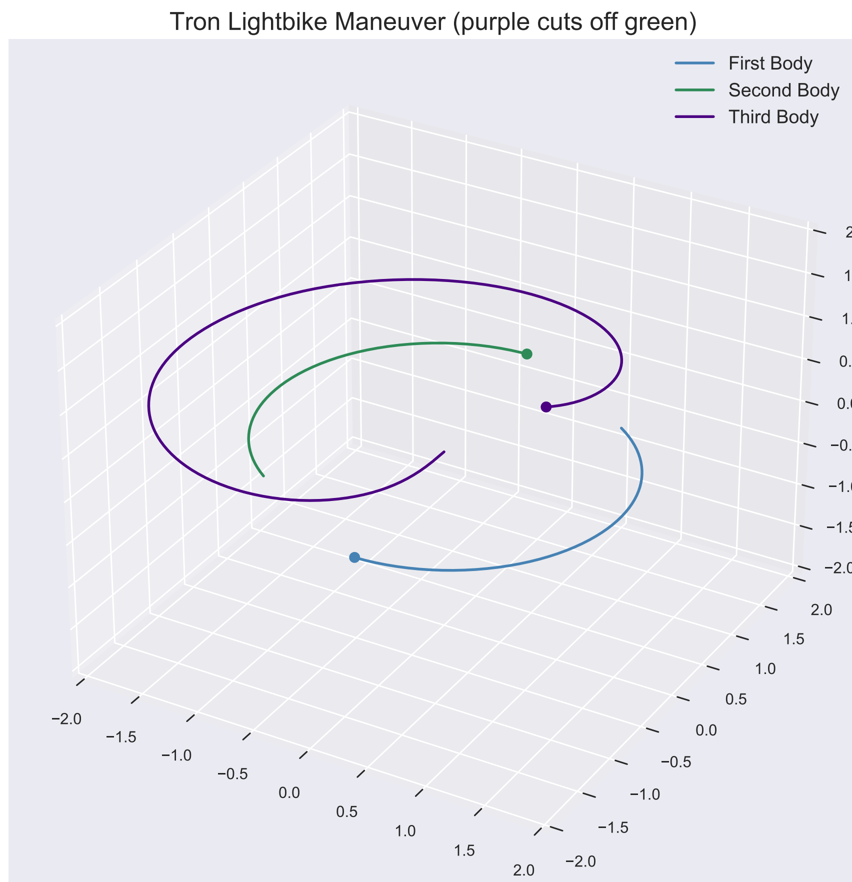

Figure 4: This set of initial conditions produce an animation that is reminiscient of a tron lightbike match, where purple cuts off green.

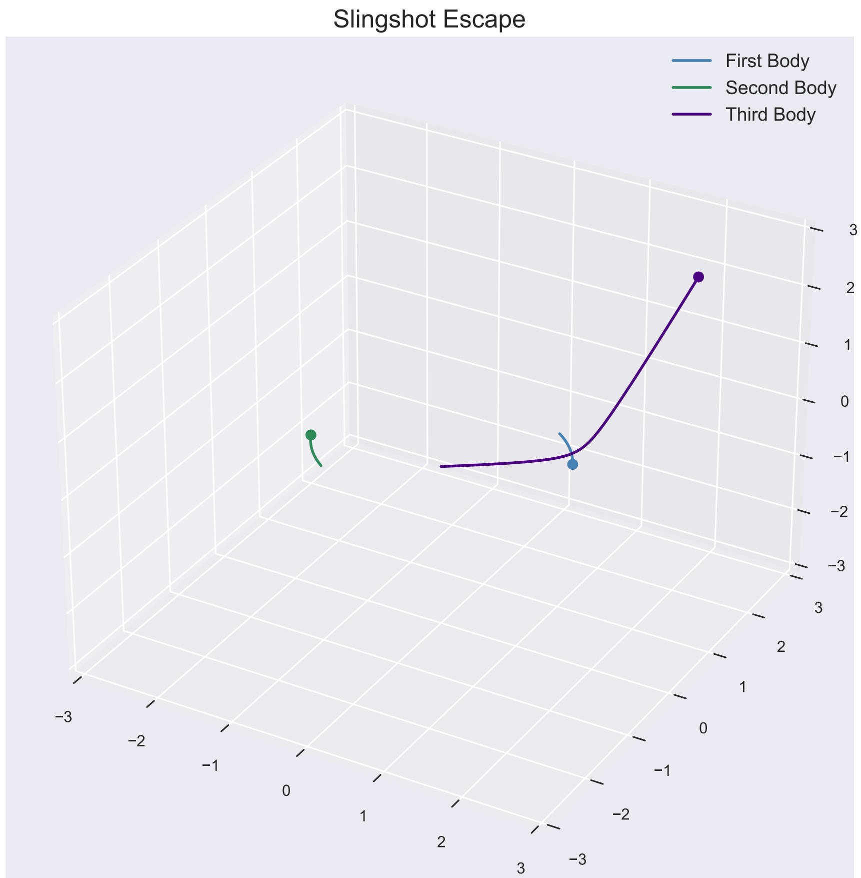

Figure 5: This was our first time finding initial conditions that allow the third body to slingshot off one of the primaries and then escape the system.

This is the closest we could get to a successful slingshot maneuver on this first attempt. We came by this mostly by trial and error. Though we have a long way to go, we note that the third body does indeed gain velocity and change direction by passing very close to the first body, so this is a solid initial model and illustrates our goal.

We further note that this is a very simple toy problem, with the first two primary bodies having mass of unity and the third body being considered as massless, so the masses, velocities, and positions for this model are pretty much meaningless, and only give us information relative to this model.

Interestingly, we added a small z perturbation, and the trajectory was still a slingshot.

3 Slingshot Exploration

We now perform a grid search to really find an optimal slingshot trajectory.

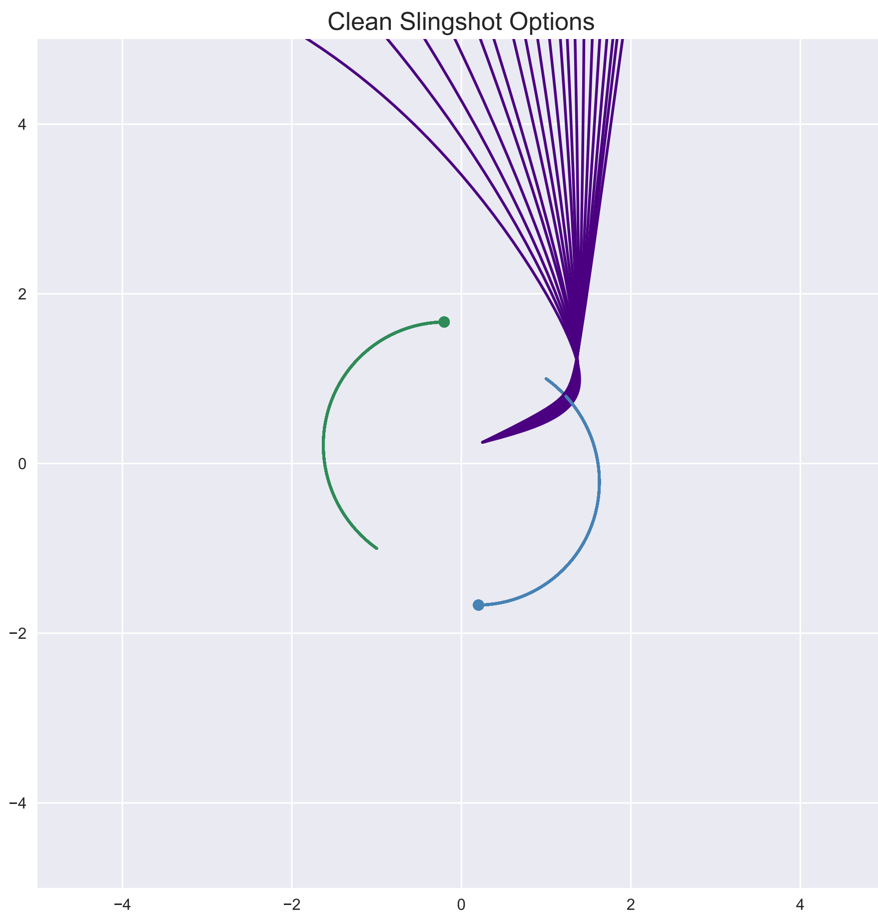

Figure 6: Here we overlay the plots of several viable slingshot maneuvers for the third body (note that we are using a toy problem with equal mass primaries here). As we can see, slight changes in the initial velocities result in significant changes in the end trajectory, but with the scale of velocities we are using, we at least get consistent, predictable trajectories in the plane.

As we can see, with these higher starting velocities, the problem appears quite stable. See the plot below however to note the chaotic behavior that appears with lower starting velocities. If the velocity is not great enough, slight changes to initial conditions result in drastically different behavior, with some slingshotting down as we would like, and others looping around unpredictably. This is perhaps something to keep in mind as we proceed, especially as we consider what realistic velocities for spacecraft are relative to the planetary bodies. We might need the model to be more robust to lower velocities, depending on what the engineering constraints are for realistic spacecraft speeds.

An animation of this plot can be found on our GitHub (link at the top).

Figure 7: We again overlay the plots of several slingshot maneuvers with slightly different initial velocities, but in this case, the velocities are too small. As a result, the trajectories we get can be chaotic, changing drastically even with only slightly different initial conditions. We note that, as seen in the animation above, this chaotic behavior is tamed by using greater starting velocities for the third body, making it much less subject to chaotic oscillations caused by the primaries since it escapes sooner and more reliably.

4 Energy Calculation

One of the interesting effects of the gravitational slingshot is that the mass which is slingshotted gains energy. To be able to explore and quantify this effect, we decided that we would calculate the energies of the masses over time. Suppose that the total energy is given by E. Then we can perform a dimensional scaling (similar to the one described in the introductory section) to get a dimensionless form of the energy ε. This is obtained through the scaling $\varepsilon = E \cdot L/(GM^2) = E \cdot \Theta^2/(ML^2)$. Using this dimensionless energy, the potential and kinetic energies are

$$\varepsilon_K(\tau) = \frac{1}{2}\sum_{i=1}^{n} \mu_i \|\mathbf{v}_i(\tau)\|^2, \qquad \varepsilon_P(\tau) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j\neq i} \frac{\mu_i \mu_j}{\|\mathbf{u}_i(\tau) - \mathbf{u}_j(\tau)\|}.$$

Newton's laws predict that the total energy should stay constant over time. To analyze a slingshot, we found it more useful to look at the mechanical energy (the sum of potential and kinetic energies) for each individual body, but because the initial model we used has the third body massless ($m_3 = 0$), the mechanical energy of this body would technically be zero. To overcome this, we calculated the ratio of energy to mass for each body using the formula

$$e_i(\tau) = \frac{1}{2}\|\mathbf{v}_i(\tau)\|^2 - \sum_{j \neq i} \frac{\mu_j}{\|\mathbf{u}_i(\tau) - \mathbf{u}_j(\tau)\|}.$$

a

5 Rotational Coordinates

In most situations where we want the slingshot effect to occur, the first two bodies are orbiting each other on a relatively stable elliptical (almost circular) orbit. This suggests that we might be able to improve the visualizations of our model by displaying it in the rotating frame of the first two bodies. So we added code allowing us to plot the solution in rotational coordinates. This code is found on our GitHub in `Code/plotting.py`.

In order to determine the rate of rotation of the frame of the first two bodies, we have to do a little math. According to [1], as long as the third mass is zero (which it is, approximately) the motion of the first and second bodies is modeled by the Kepler problem. In the ideal situation, the two bodies move in circular orbits about their center of mass with a frequency

$$\Omega = \sqrt{\frac{G(M_1 + M_2)}{R^3}},$$

where R is the distance between them. We can perform a coordinate change to put the positions, velocities, and accelerations of the bodies into a rotating frame centered at the center of mass of the system and rotating with angular velocity Ω. In this coordinate system the first two bodies M_1 and M_2 stay at rest (their velocities v_1 and v_2 are canceled out by rotating the coordinate system by subtracting Ωr_1 and Ωr_2).

In practice, we can use the change in the solution for the first body's movement to find how much to rotate the coordinates, construct a rotation matrix, and use the rotation matrix to modify the position of the third body. Supposing we have already found the position of the center of mass of the system and subtracted off its position so our coordinates are already centered around the origin, we can find the rotation θ of the first body over time (compared to its initial position) by using a 2-argument arctangent: $\theta(t_i) = \arctan(y_1(t_i) - y_1(t_0), x_1(t_i) - x_1(t_0)$; in numpy `arctan2(y,x)`. (Using 2 arguments allows us to determine the correct quandrant for θ). Now we will set the first two bodies positions constant based on their initial positions and rotate the third body by $-\theta(t_i)$ for each time step. This can be accomplished nicely by matrix multiplying a rotational matrix based off $-\theta_i = -\theta(t_i)$ to the x,y position vector of the third body at each time step:

$$R(-\theta_i) = \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix}$$

Because we are rotating our coordinates around the origin, we can use the same rotation matrix to convert our velocities into rotating coordinate velocities. If the center of mass has a velocity, we must subtract off that velocity from every bodies' velocity before rotating.

Below is a plot with the energies, as well as a plot of the solution using rotating coordinates with respect to the first two bodies.

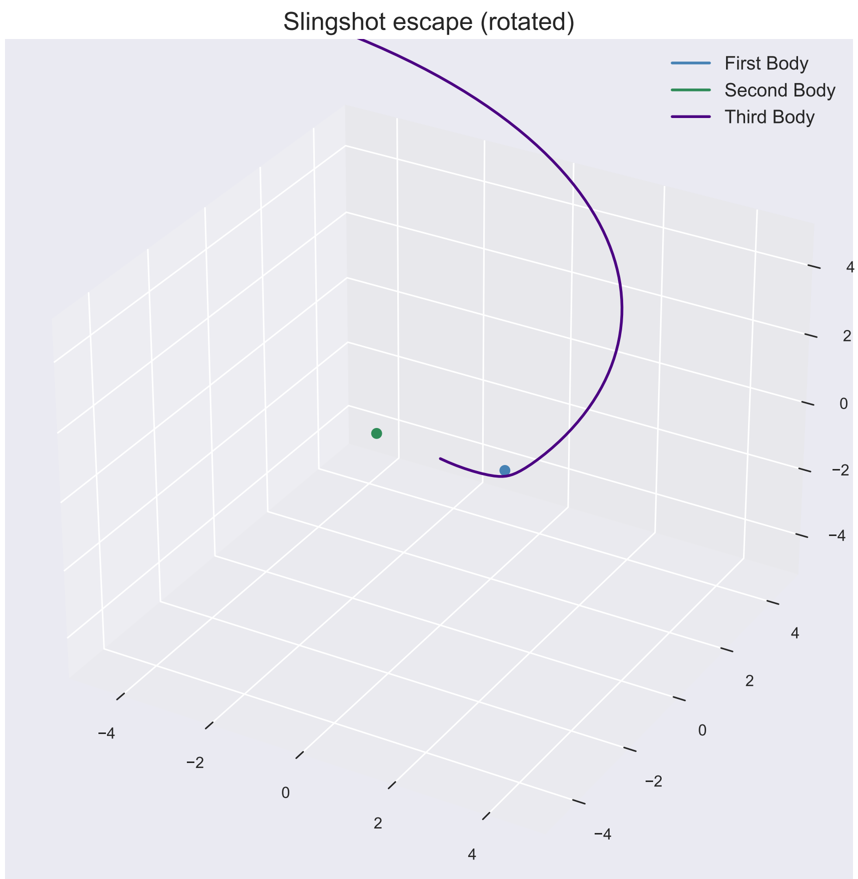

Figure 8: Here we have plotted our successful, escaped slingshot, now in rotating coordinates. That is, we shift our frame of reference with the primaries so that they appear to remain stationary while the third body moves around them.

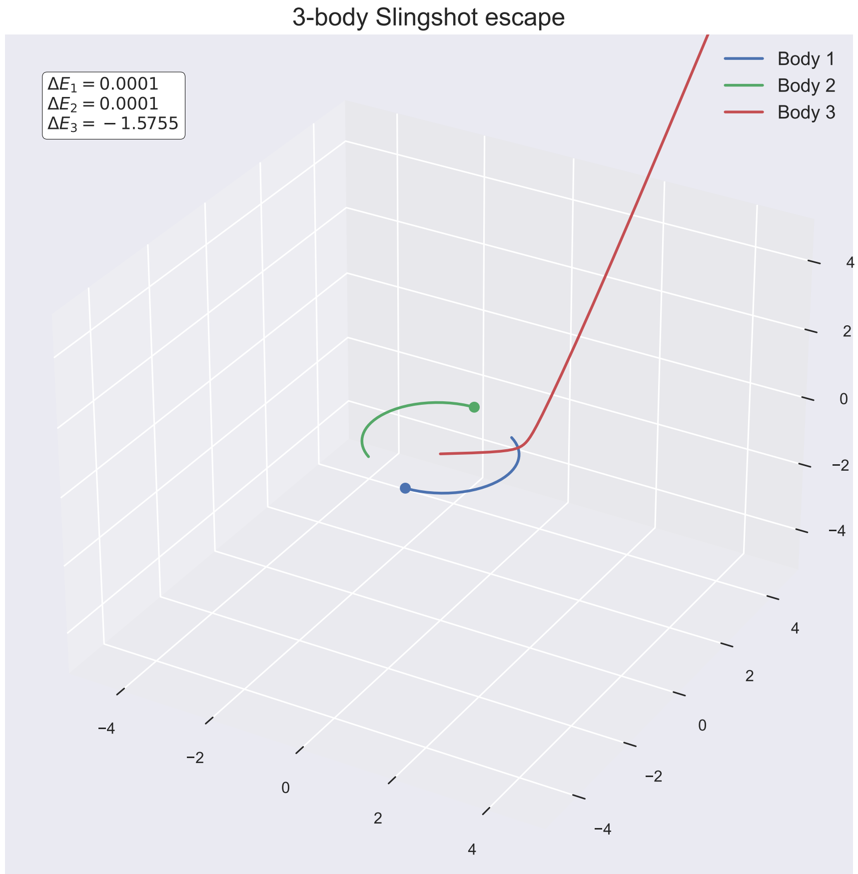

Figure 9: Here, we plot the same escape trajectory as before, but with one significant change. Now, we have started the 3rd body's postion along the -z axis and gave it a positive z velocity. The model seems to be robust to this change, and the results are comfortingly similar to those from before.

6 Making the Model More Realistic

This is still a toy problem, but a slightly more realistic one. Instead of equal masses set to 1, we have a large mass corresponding to the sun and a smaller mass corresponding to the planet to be used in the slingshot maneuver.

We were able to pull off a decent slingshot here, and we can likely further improve it in the future. Though this is still a toy problem, with made up masses and initial conditions, it is much more realistic than the previous model. We now have a large mass to represent the sun, and a smaller mass (but still very large relative to the "massless" spacecraft) to represent Jupiter, or any planet that we could attempt a slingshot with.

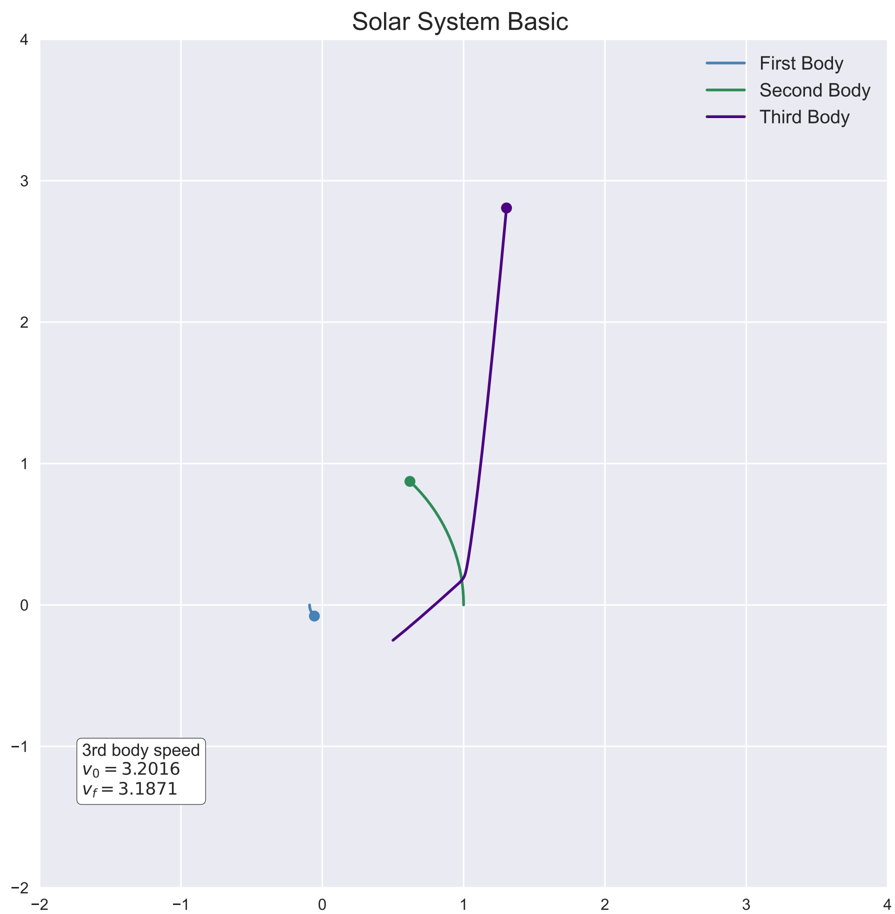

Figure 10: This model, like the previous ones, assumes that the third body is massless, but now we are using with the proper mass ratio to model the interaction between the sun and Jupiter. As we can see, we are able to pull off a successful slingshot maneuver.

If you view the animation of this model, we see that the spacecraft does indeed gain significant velocity as it passes by the second body, indicating that our slingshot model is indeed working as desired. We will continue to engineer it so that we can maximize the velocity gain.

We also constructed a version of the model with the third body having a small positive (but still nonzero) mass.

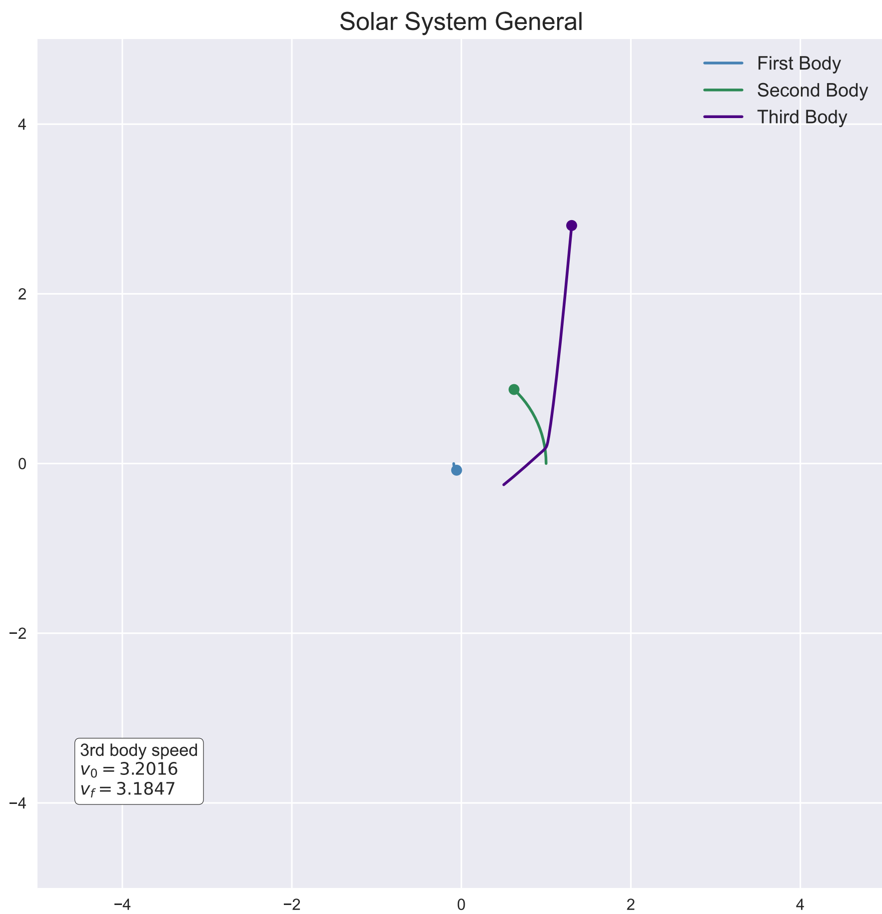

Figure 11: This model also uses the proper mass ratio for the sun and Jupiter, rather than the toy problems in other animations. Now, however we give the third body a small positive mass (rather than modeling it as being massless). This is slightly more realistic, but encouragingly, the results are basically identical to the previous model that used a massless third body, indicating that our model is structurally stable.

We can compare the ”basic” model (with a massless third body) with the ”general” model (where the third body has a small positive mass).

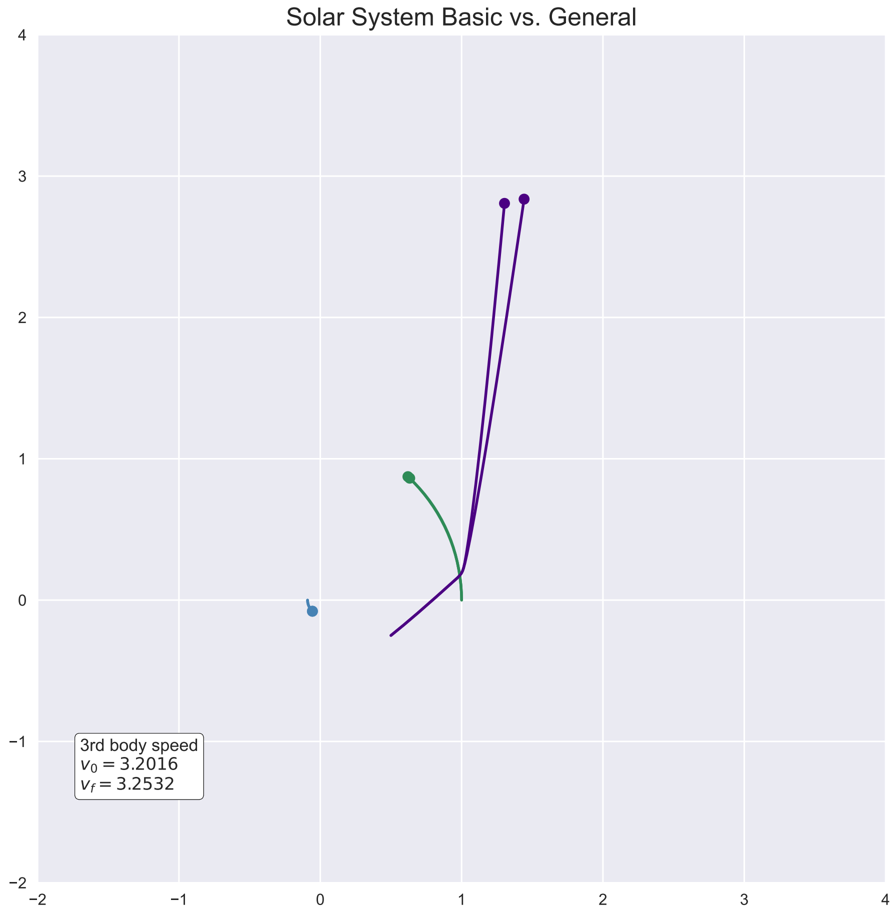

Figure 12: Here we plot the difference between a zero mass third body model and a small positive mass third body model.

We can also perform a grid search to find the best slingshot in this model:

Figure 13: Here we overlay the animations of several viable slingshot maneuvers in the realistic solar system model. As we can see, slight changes in the initial velocities result in significant changes in the end trajectory, but with the scale of velocities we are using, we at least get consistent, predictable trajectories in the plane.

7 Further Exploration

One topic that we would like to explore further is how to find initial conditions which result in a trajectory with a specific angle following a slingshot. To do this we think it would be a good idea to perform a grid search to find the right initial conditions.

We also would like to find slingshots which give a maximum energy gain. To do this we would use a grid search as well.

References

[1] S. Widnall, ”Lecture L18 - Exploring the Neighborhood: the Restricted Three-Body Problem.”

https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-07-dynamics-fall-2009/lecture-notes/MIT16_07F09_Lec18.pdf