Modeling the Gravitational Slingshot Effect

Sam Cochran, Jacob Murri, and Caleb Wilson

December 11, 2020

Abstract

We attempt to model the physics of the gravitational slingshot effect, whereby a satellite can swing around a planet and receive a boost in speed so that it can successfully reach its destination. Beginning with a system of ordinary differential equations derived from Newton's Laws, we code up a numerical N-body problem solver. We determine which assumptions are necessary to produce a workable model, and implement a method to obtain a slingshot trajectory passing through a specific point by casting the problem in terms of optimizing an objective function. Our code and animations can be viewed **here**.

1 Background and Motivation

In order to explore the universe, mankind has sought to send satellites all over the solar system. One technique that is frequently valuable in getting these satellites where they need to be is the gravitational slingshot effect, where a satellite's trajectory is fine-tuned to allow it to pass close by a planet so that it can get a velocity boost without the use of fuel. A helpful analogy is that of bouncing a ball off of a moving train. The collision is elastic, and so momentum is conserved; however, the train has a much larger mass than the ball, so the effect of the collision on the train is negligible, while the ball may receive a significant boost in speed. Similarly, a satellite passing by a moving planet can receive a large velocity boost that will enable it to reach its destination. Our goal in this project is to devise a method for choosing initial velocity conditions for a satellite that result in a slingshot maneuver past a large planet and toward a desired 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

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

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 introducing the dimensionless parameters $\mathbf{u}_i = \mathbf{x}_i / L$, $\tau = t/\Theta$, and $\mu_i = m_i / M$ allows us to rewrite the equations as

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

Then by choosing our mass, length, and time scales appropriately, we can set $G M \Theta^2/L^3 = 1$, 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

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

Now these equations are in the form we need to be able to write a numerical solver. A description and analysis of our modeling efforts is provided in section 2.

2 Modeling and Analysis

Because we developed our model and found our slingshot in an iterative fashion, we felt it was best to describe the model and analyze it iteratively, rather than presenting the model and the analysis in separate sections.

2.1 Initial Models

In the gravitational slingshot maneuver, the mass of the satellite is insignificant compared to the mass of the planet it swings around. 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). Specifically, to explore how well our code can model the slingshot effect, we choose a setup where $m_1 = 1, m_2 = 1, m_3 = 0$.

Given the system (2), the masses listed above, and initial conditions (comprising initial positions and initial velocities), we can numerically integrate to determine the trajectories over time. The code that performs the numerical integration is found on **our GitHub** [1] in the file `Code/simulation.py`. Figures 1-4 represent our first (manual) attempts at finding initial conditions which yield a slingshot. They illustrate, more than anything, the difficulty of choosing initial conditions that allow for a successful slingshot maneuver. Our task moving forward was to find a more robust way to choose good initial conditions that allow for successful slingshot maneuvers. We eventually came up with a good solution that uses an automated grid search (see section 2.6).

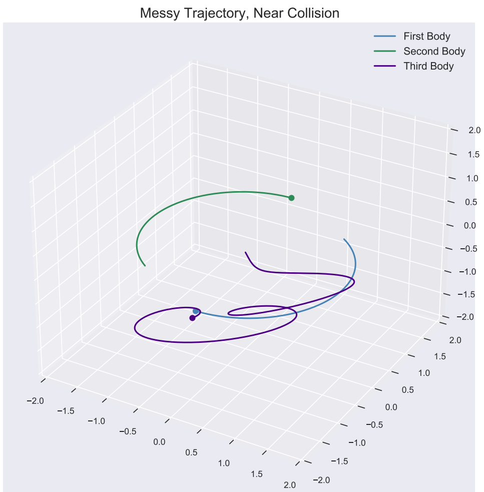

Figure 1: This plot shows a messy trajectory achieved by choosing specific initial conditions. It is interesting to watch how the third body moves in tandem with one of the larger primaries, oscillating around it as both orbit the barycenter.

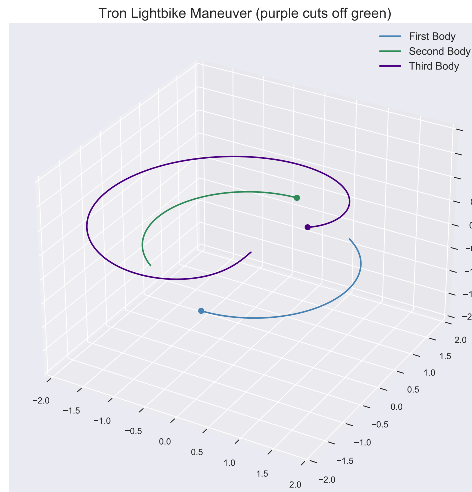

Figure 2: This set of initial conditions produce an animation that is reminiscent of a Tron lightbike match, where purple cuts off green.

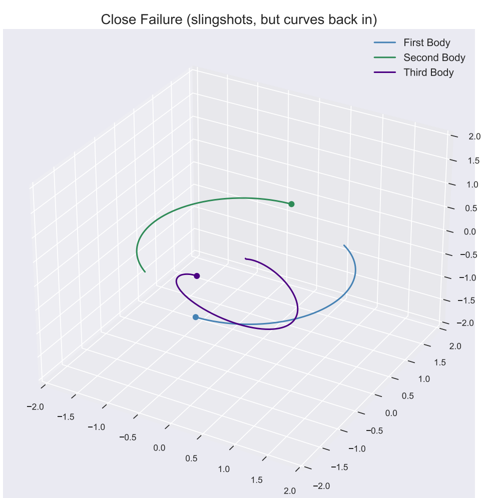

Figure 3: Here, we finally found a set of initial conditions that came close to allowing for a successful slingshot, though the third body curves back in. Still, it does gain a lot of velocity through the maneuver, which is what we are trying to capture in our model.

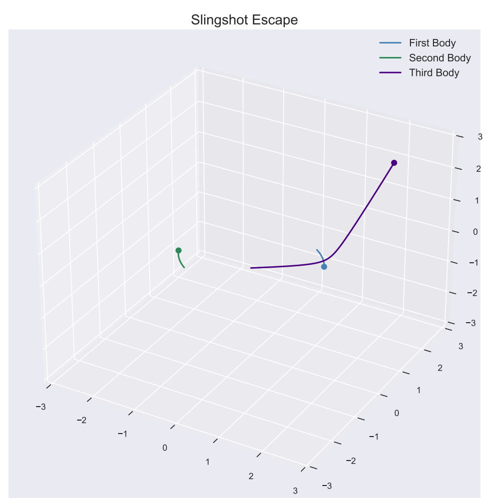

Figure 4: This was our first time finding initial conditions that allow the third body to slingshot off one of the primaries and then escape the system.

This is the closest we could get to a successful slingshot maneuver on this first attempt. We came up with these initial conditions using 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 1 and the third body being massless, so the masses, velocities, and positions predicted by this model do not correspond to actual solar system orbits, and are essentially meaningless without scaling (the reverse of the operations we performed in the introduction). Interestingly, we tried adding a small z perturbation, and the trajectory was still a slingshot, so the structure of our model is stable to nonzero motion in the z

2.2 Finding Our First Good Slingshots

We now perform a grid search to find an optimal slingshot trajectory. As we can see from Figure 5, the problem appears quite stable when using higher starting velocities. But when we give the satellite smaller starting velocities, chaotic behavior becomes evident, as seen in Figure 6. If the velocity is not large enough for the 3rd body to escape, slight changes to initial conditions result in drastically different behavior, with some slingshotting the satellite as we might like, and others looping around unpredictably. This is 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.

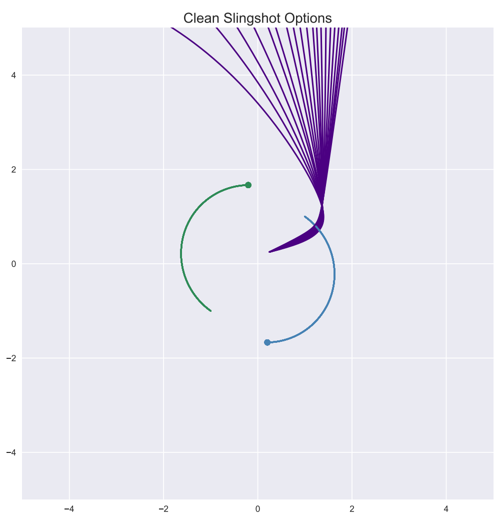

Figure 5: Here we overlay the plots of several viable slingshot maneuvers for the third body (note that we are using a toy problem with equal mass primaries here). As we can see, slight changes in the initial velocities result in significant changes in the end trajectory, but with the scale of velocities we are using, we at least get consistent, predictable trajectories in the plane.

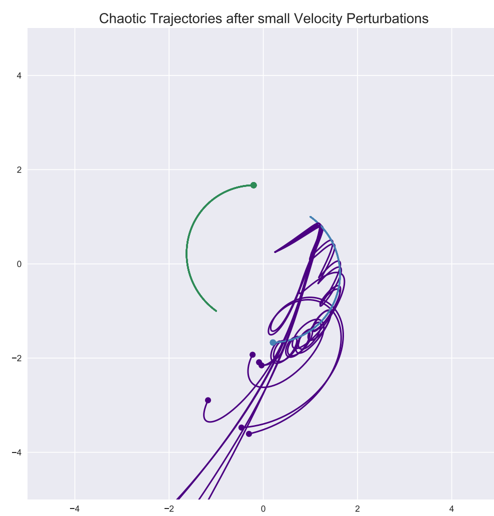

Figure 6: We again overlay the plots of several slingshot maneuvers with slightly different initial velocities, but in this case, the velocities are too small. As a result, the trajectories we get can be chaotic, changing drastically even with only slightly different initial conditions.

An animation of these plots can be found on **our GitHub** [1].

2.3 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 mechanical energy (the sum of the kinetic and potential energy) should stay constant over time. We calculated the energy over time for our model and the result showed that our model mostly conserves energy, but due to numerical error the energy slowly decreases over time. This means that our model is not very useful for predicting long-term behavior of a system, but works just fine for short-term behavior like a slingshot.

As mentioned earlier, we expected to see that the slingshot maneuver would increase the mechanical energy of the third body. But we realized that because the model we used often has a massless third body, ($m_3 = 0$), 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

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

and our analysis showed that the energy of the third body often increased after a good slingshot.

2.4 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** [1] in `Code/plotting.py`.

According to [2], as long as the third mass is approximately zero relative to the mass of the primaries, 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 displacement of the first body over the first few time steps to find how much to rotate the coordinates. Then we can construct a rotation matrix and use it to modify the position of the third body. Supposing we have already found the position of the center of mass of the system and centered our coordinates around it, we can find the rotation θ of the first body over time by using a 2-argument arctangent: $\theta(t_i) = \arctan\left(y_1(t_i) - y_1(t_0), x_1(t_i) - x_1(t_0)\right)$. 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 using the following rotational matrix, where $-\theta_i = -\theta(t_i)$ is the angle 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 nonzero velocity, we must subtract that velocity from the velocity of each of the other bodies before rotating.

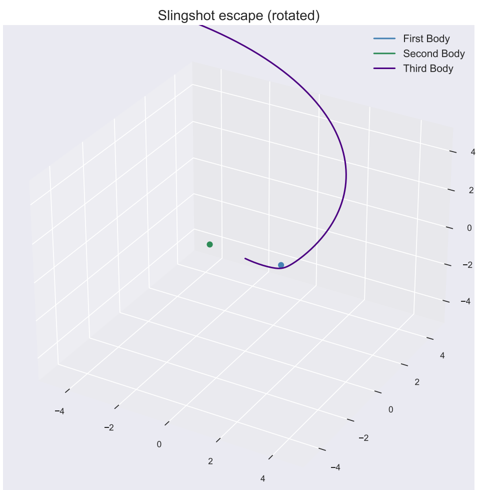

Figure 7: Here we have plotted our successful, escaped slingshot, now in rotating coordinates. That is, we shift our frame of reference with the primaries so that they appear to remain stationary while the third body moves around them.

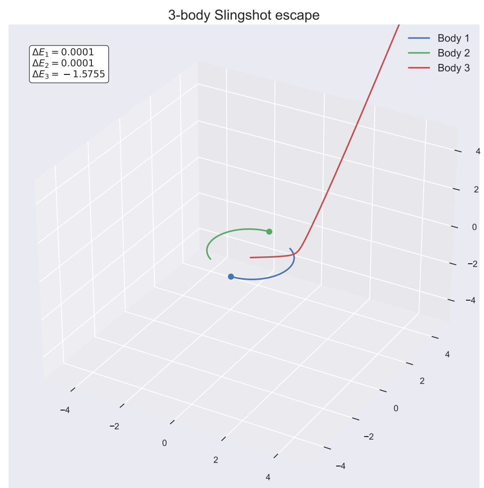

Figure 8: Here, we plot the same escape trajectory as before, but with one significant change. Now, we have started the 3rd body's position along the negative z axis and given it a positive z velocity. The model seems to be robust to this change, and the results are comfortingly similar to those from before.

2.5 Modeling a System with Realistic Masses

Now, we move on from modeling the equal-mass toy problem above to modeling the dynamics of our solar system using more realistic mass ratios. Instead of equal masses, we have a large mass corresponding to the Sun and a smaller mass corresponding to the planet to be used in the slingshot maneuver, with the mass ratio proportional to the actual mass ratio between the Sun and Jupiter. We begin by continuing with our third body having zero mass, and then expand our model to handle the case where the third body has a small positive mass.

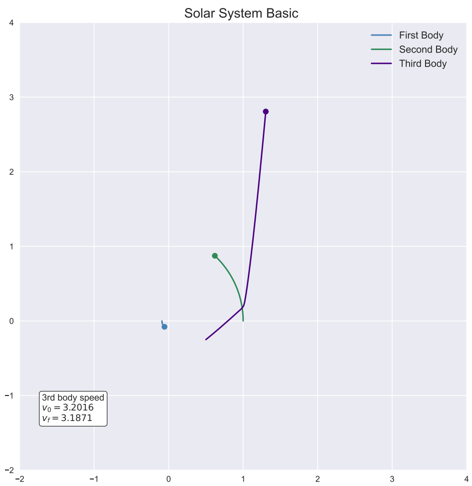

Figure 9: This model, like the previous ones, assumes that the third body is massless, but now we are using with the proper mass ratio to model the interaction between the sun and Jupiter. As we can see, we are able to pull off a successful slingshot maneuver.

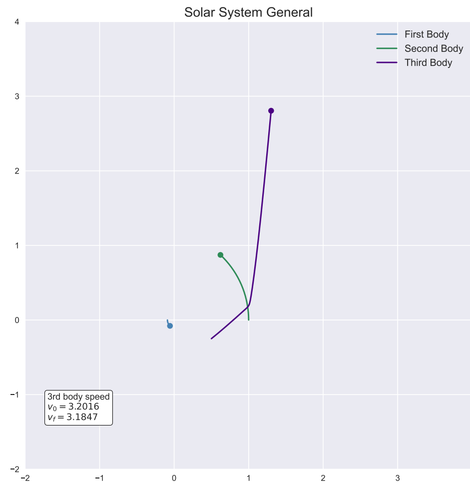

Figure 10: Now we give the third body a small positive mass (rather than modeling it as being massless). This is slightly more realistic, and encouragingly, the results are basically identical to the previous model that used a massless third body, indicating that our model is structurally stable.

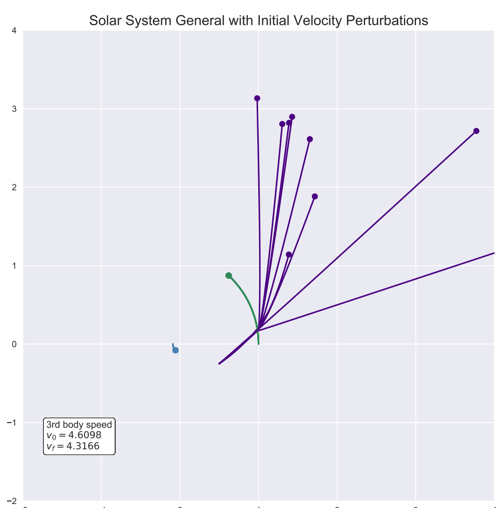

Figure 11: Here we overlay the animations of several viable slingshot maneuvers in the realistic solar system model, each corresponding to a slightly different initial velocity condition for the third body. As we can see, slight changes in the initial velocities result in significant changes in the end trajectory, but with the scale of velocities we are using, we at least get consistent, predictable trajectories in the plane.

2.6 Automating the Search for Slingshot-Producing Initial Conditions

An important part of our project was developing a method to find slingshot trajectories that allow the satellite to reach or pass through a given point. Once we had a function that could accurately predict the trajectory of the bodies given initial conditions, we were able to automate a grid search to find which initial conditions slingshot the third body (the satellite) to a given location. We created a function that accepts a target point in space and a guess for a set of initial conditions, and finds initial velocity conditions for the third body that allow it to pass through the goal point.

The basic idea behind this function is to cast the problem in terms of optimization. We create an objective function that accepts initial velocity conditions for the third body and returns the distance between the target point and the closest point to it along the resulting trajectory. Minimizing this objective allows us to find a slingshot maneuver that passes through the target point by choosing the right initial velocity conditions for the third body.

Amazingly, we are able to execute a wide array of very specific trajectories by changing only the initial velocity conditions for the third body. Using these optimization tools, we are able to very accurately get the third body to pass through the point we want it to by starting it with the right velocity (see Figures 12, 13, and 14). Thus, as long as we can place the third body in the indicated position in space with the proper velocity, it will arrive at the desired destination without having to use any power or thrust.

It should be noted that this grid search method is still limited in what it can do. For now, it is only set up to find trajectories that lie in the plane. Additionally, this method is not guaranteed to work if the desired point is too far outside of a certain feasibility range based on the satellite's initial position and the initial velocity guess passed into the function. If it is given a bad guess for the initial velocity conditions, the function may return conditions that do not even produce a slingshot trajectory. It is also expensive to run, since each iterative step in the optimization process must run the entire simulation to produce a trajectory. Thus, there are still many improvements to be made.

Still, under the right circumstances this method is very effective. Good practice is to use trial and error (like we did in earlier sections) to find initial conditions that result in a decent slingshot maneuver in the right general direction. Then, using these conditions as our initial guess, the function is able to find conditions that result in a slingshot toward any desired point within approximately 90 degrees in either direction from the guess trajectory (measuring angle from the point of slingshot).

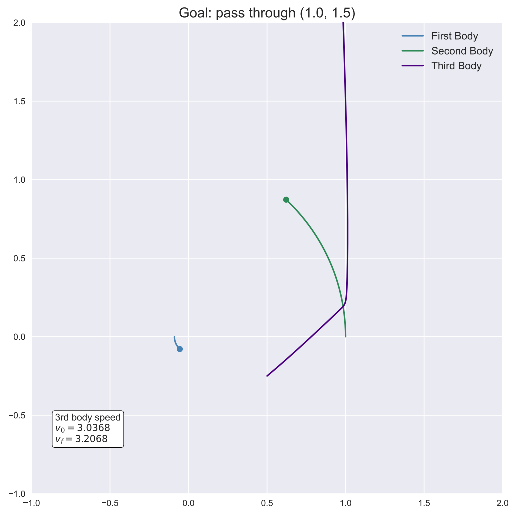

Figure 12: A slingshot maneuver optimized to pass through the point $(x, y) = (1, 1.5)$.

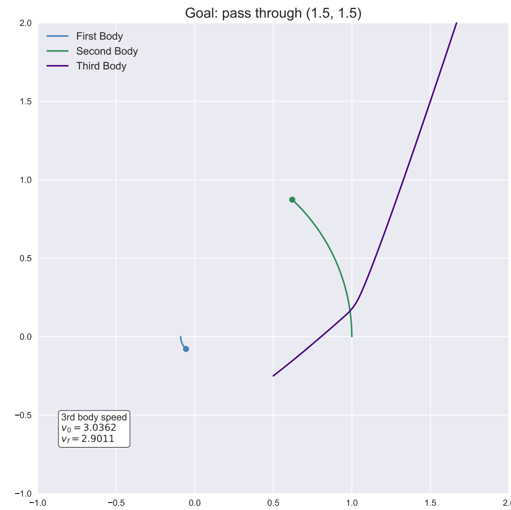

Figure 13: A slingshot maneuver optimized to pass through the point $(x, y) = (1.5, 1.5)$.

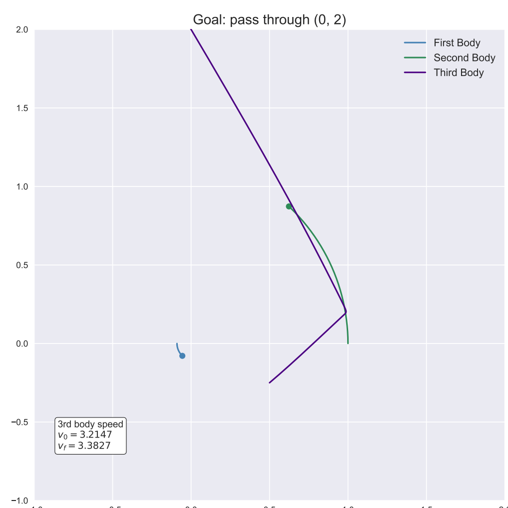

Figure 14: A slingshot maneuver optimized to pass through the point $(x, y) = (0, 2)$.

3 Results and Conclusion

As we progressed through the various iterations of our model, we were struck by how well they seemed to work. As we moved from a simple problem to more complex ones, Newton's laws continued to provide solid models that were robust to changes and additions. We went from modeling a toy problem with equal mass primaries and a zero mass third body to modeling a three body problem with accurate mass ratios representing the sun, Jupiter, and a satellite with a small positive mass, and our model was quite robust to these changes. As we added features to represent these more complicated dynamics, we observed only small changes from our original problem where the third body was restricted to have zero mass.

Notably, we didn't seem to have much of an issue with numerical error, which is something we were worried about at the outset of the project. We would have to compare our model to other models, or run real-world tests to fully confirm, but we expect that our model does a decent job of approximating reality.

We would also note that the predictions that the model makes are usually dependable and not chaotic, but when the bodies pass very close to each other the trajectories can change a lot, implying that there is some chaos in the system. Because of this, we wouldn't want to use our model to predict a long-term trajectory or orbit because we know that the orbits decay. But it seems good enough to predict over the length of time required for a slingshot in a space mission, which was always the intended purpose of the model.

In summary, the 3-body model is a good model for predicting gravitational slingshots. We have discovered basically what NASA already knew and, in a small measure, validated their approach.

3.1 Future Work

One topic that we would like to explore in the future is which trajectories will maximize the energy gained from a slingshot. We have been fairly successful at finding initial conditions which produce reliable slingshots to a target point, but we have not tried to discover which initial conditions lead to maximum energy gain. The code we have already written to calculate the energy would be extremely helpful, and we expect that we could use a grid search much like the one described in section 2.6.

Some of our peers have suggested that we could improve our model by implementing relativistic corrections to the equations of motion. We agree that this would be an interesting exercise, however it would take far too long to implement before the submission of this project. Furthermore, because satellites travel at speeds which are small compared to the speed of light and the masses of objects in the solar system are small compared to the masses of black holes (and our 3rd body does not get too close to other masses), we expect that neither special nor general relativity would make a significant difference in our predictions.

Another idea we could implement would be finding slingshot trajectories in n-body systems with n greater than 3. We already have the code to model general n-body systems, so our work would just be to use that to find trajectories that result in slingshot maneuvers in systems with 4 or more bodies.

References

[1] S. Cochran, J. Murri, and C. Wilson. https://samcochran.github.io/Gravitational-Slingshot/

[2] 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

[3] J. Marsden, S. Ross, "New Methods in Celestial Mechanics and Mission Design." http://www.cds.caltech.edu/~marsden/bib/2006/01-MaRo2006/MaRo2006.pdf