Halo Orbit Simulation of JWST at L_2

CSE 6730 Project - Final Report

December 3, 2025

Group 15

Tianyang Hu (thu98@gatech.edu)

Haoran Yan (hyan82@gatech.edu)

Mark Zhang (markzhang@gatech.edu)

Shiqi Fan (sfan84@gatech.edu)

GitHub Repository

https://github.com/TimHu-0728/CSE-6730-Final-Project

Summary Video

https://mediaspace.gatech.edu/media/%5BTeam+15%5D+Halo+Orbit+Simulation+of+JWST+at+L2.mov/1_sv9vhsje/394668913

Abstract

The James Webb Space Telescope (JWST) operates in a large halo orbit around the Sun–Earth L_2 Lagrange point, where the combined gravity of the Sun and Earth creates a favorable environment for continuous deep-space observation. In this project, we use the Circular Restricted Three-Body Problem (CR3BP) to model JWST's motion. Our goals are to design a transfer trajectory from Earth to the L_2 region and to study the spacecraft's motion once it is on a halo orbit.

Within the non-dimensional Sun–Earth CR3BP framework, we solve the equations of motion to obtain a three-dimensional halo orbit around L_2, and then construct an optimized transfer trajectory that brings the spacecraft from an Earth-centered parking orbit to the vicinity of this halo orbit. A key contribution of our work is the explicit modeling of the transfer phase together with a 3D visualization gallery, where we render the transfer and halo motion to provide an intuitive view of JWST's trajectory in the rotating Sun–Earth frame.

We validate our simulator using face, analytical-model, and behavior validation. Visually, the simulated LEO-to-halo trajectory exhibits the expected three-dimensional, inclined halo geometry in the rotating frame. Analytically, the Jacobi constant is preserved to within numerical roundoff (on the order of 10^{-12}) over a two-year window. Behaviorally, the distance from the spacecraft to L_2 oscillates between about $0.80r_0$ and $1.00r_0$ with no long-term drift, indicating a bounded halo orbit. Together, these steps provide an end-to-end simulation pipeline—from departure near Earth to stable halo motion around L_2—that captures the essential behavior of a JWST-like mission in the Sun–Earth CR3BP model.

1 Project description

1.1 Project Goals

The goal of this project is to build and validate a numerical model of the James Webb Space Telescope (JWST) in the Sun–Earth system using the Circular Restricted Three-Body Problem (CR3BP) (11). We focus on two main tasks that together form an end-to-end simulation pipeline:

- **Transfer to L_2:** We design and simulate a transfer trajectory that carries a JWST-like spacecraft from an Earth-centered parking orbit to the region around the Sun–Earth L_2 Lagrange point. We start with a spacecraft at Earth's Lower Earth Orbit and choose its input $u(t)$ so that, under the CR3BP dynamics, it gradually travels into the vicinity of the desired halo orbit. This explicit modeling of the transfer phase is one of the innovative parts of our project, since many CR3BP examples focus only on the halo orbit itself and not on how the spacecraft gets there.

- **Halo orbit around L_2:** Once the spacecraft is near L_2, we simulate its motion on a three-dimensional halo orbit by solving the non-dimensional equations of motion in the rotating Sun–Earth frame. We then build a 3D visualization gallery using `PyVista`, which includes rendered animations of both the transfer and the halo orbit from multiple viewpoints. This gallery helps us and future readers to see how the trajectory evolves in space and how the halo orbit sits relative to the Sun and Earth, and it represents another distinctive feature of our project beyond basic time-series plots.

We validate our simulator using a combination of *face validation*, *analytical-model validation*, and *behavior validation.* For face validation, we visually inspect the simulated LEO-to-halo trajectory in the rotating frame and confirm that the transfer and final orbit exhibit the characteristic three-dimensional, inclined halo geometry seen in reference Sun–Earth L_2 trajectories. For analytical-model validation, we monitor the Jacobi constant along the simulated path and observe that its drift $C(t) - C(0)$ remains on the order of 10^{-12} over two years (Figure 4), consistent with numerical roundoff for our Runge–Kutta integrator. For behavior validation, we track the distance from the spacecraft to L_2 and find that it oscillates periodically between about $0.80r_0$ and $1.00r_0$ with no long-term growth or decay indicating a bounded halo orbit. Together, these checks provide complementary evidence that our CR3BP simulator reproduces the expected Sun–Earth L_2 dynamics and forms a credible basis for the trajectory design experiments that follow.

1.2 Physical Phenomenon and Modeling Choices

JWST does not orbit the Earth in a simple circle like a low-Earth-orbit satellite. Instead, it operates near the Sun–Earth L_2 Lagrange point, a special location where the combined gravity of the Sun and Earth and the apparent centrifugal force balance in a frame that rotates with the Earth around the Sun. The L_2 point lies on the line from the Sun through the Earth, slightly beyond Earth's orbit. Near this point, a small object like JWST can "hover" relative to the Earth and Sun with modest fuel usage, while keeping its sunshield pointed toward both bodies and maintaining a continuous view of deep space.

In reality, JWST does not sit exactly at L_2. It moves on a three-dimensional loop around it, called a halo orbit, which appears as a twisted path that passes above and below the plane of Earth's orbit when viewed in the rotating frame. Capturing the main shape, period, and qualitative stability of this halo orbit is a central goal of our model. To describe this behavior, we adopt the Circular Restricted Three-Body Problem (CR3BP) as following:

- The Sun and Earth are treated as two massive bodies moving on circular orbits around their common center of mass, while JWST is a third body with negligible mass that feels their gravity but does not influence their motion.

- The equations of motion are written in the rotating frame where the Sun and Earth remain fixed, which makes the geometry of the L_2 region and the halo orbit easier to understand.

- The system is non-dimensionalized by scaling distances, masses, and time so that the main quantities are of order one, improving numerical stability and simplifying parameter values.

- An energy-like invariant, the Jacobi constant, is monitored to verify that the numerical integration remains consistent with the underlying CR3BP dynamics.

At the same time, we deliberately simplify the problem by ignoring several real-world effects: we do not include solar radiation pressure, the small eccentricity and inclination of Earth's orbit, or the detailed thrust profile of JWST's station-keeping maneuvers. These effects are important for an operational mission, but they are not necessary for capturing the essential geometry, transfer behavior, and qualitative stability of a JWST-like halo orbit at the level of this course project.

1.3 Orbit Illustration

To give an intuitive picture of the geometry we study, we include a schematic of JWST's orbit near the Sun–Earth L_2 Lagrange point. The figure shows the Sun, Earth, the location of L_2, and the large halo-like orbit that JWST follows around L_2 in the Sun–Earth system.

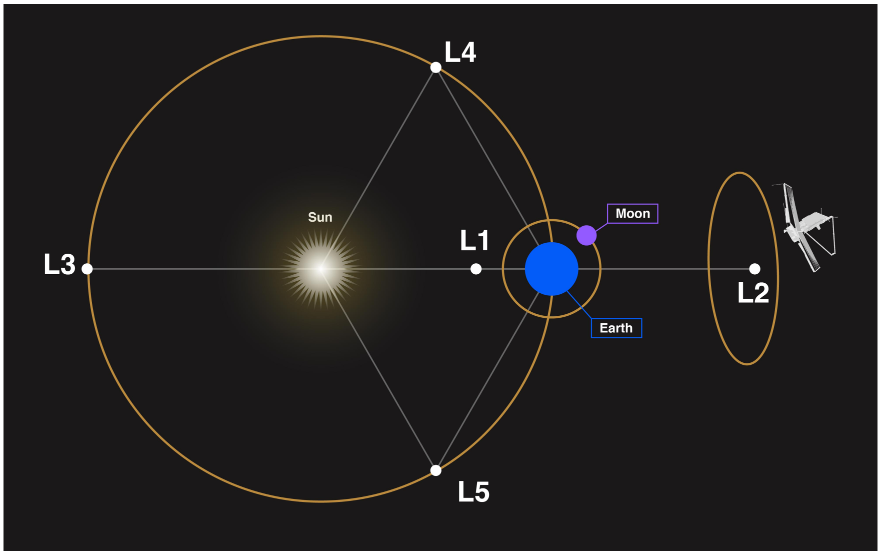


Figure 1: JWST's orbit around the Sun–Earth L_2 point. Image credit: NASA (4).

2 Literature review

The James Webb Space Telescope (JWST) operates near the Sun–Earth L_2 Lagrange point on a large halo-like orbit rather than in a low-Earth orbit. NASA's description of Webb's orbit explains that this configuration keeps JWST roughly in line with the Earth as it orbits the Sun, allowing the telescope to maintain a continuous view of deep space while its sunshield permanently blocks the Sun, Earth, and Moon (5). This motivates our focus on the L_2 environment, but does not provide the mathematical framework required for simulation.

For the dynamical model, we adopt the Circular Restricted Three-Body Problem (CR3BP), in which the Sun and Earth are treated as two massive primaries moving on circular orbits about their common center of mass, and JWST is modeled as a third body with negligible mass. Weber's online notes on the CR3BP derive the non-dimensional equations of motion in the rotating frame, introduce the Jacobi constant, and discuss the Lagrange points, including L_2 (11). Classical texts such as Prussing and Conway (7) place this model in the broader context of astrodynamics and show how it can be used to study libration-point orbits and transfers. Building on this, our project uses the Sun–Earth CR3BP to approximate JWST's three-dimensional halo-like motion around L_2.

Beyond the local dynamics, several studies investigate transfer trajectories from Earth to libration-point orbits. Howell and co-authors consider transfers from Earth parking orbits to Sun–Earth L_1/L_2 halo orbits using CR3BP-based methods (3), while Rausch (10) and Nath and Ramanan (6) explore low-Δv and optimized transfers to halo orbits. For the JWST/NGST mission, Folta et al. discuss trajectory design strategies for an L_2 libration-point mission (1). These works show that optimal transfers to halo orbits are well studied at the research level. In our course project, we take inspiration from this literature but implement a simplified approach: we design a numerically optimized transfer within the non-dimensional CR3BP and combine it with a JWST halo orbit. Practically, we use the Python Control Systems Library, python-control (8), for the CR3BP time-domain simulations, and PyVista (9) to render and animate the trajectories, yielding an end-to-end, visually interpretable simulation of a JWST-like mission around Sun–Earth L_2.

3 Conceptual model

3.1 Overview

In order to simulate the trajectory of the JWST in the vicinity of the L_2 point, we describe its dynamic using the CR3BP model. This mathematical abstraction simplifies the 3-body problem of the solar system into a tractable system of differential equations while preserving the same gravitational structures of the Lagrange points and halo orbits relevant to the problem.

3.2 Assumptions

This mathematical model relies on a few physical assumptions. Each represents and reflects specific characteristics of this physical system.

This system is defined as two large mass primaries, $m_1,\ m_2$, and one infinitesimally small mass, m_3. In our case, the Sun (m_1) and Earth (m_2) being the most massive objects dominate the dynamics of the system. The JWST is then treated as a negligibly small mass compared to the other two. Thus, we have that $m_3 \ll m_1,\ m_2$. Another assumption is that the two primaries are in circular orbit with each other, with the center being the barycenter as will be further discussed in the following section.

3.3 Non-Dimensionized Coordinate System

As shown in Fig. 2, the two finite masses move in circular orbits about their common center of gravity COG. We define COG to be the origin of the coordinate system, x-axis be parallel to the separation r_{12} pointing from the COG to m_2, and y-axis perpendicular to x-axis in the orbit plane. Lastly, assume this frame is rotating about the z-axis at constant rate ω relative to a stationary frame, the universe. The system is normalized by letting the sum of the masses be 1, the constant separation between the two major bodies r_{12} be 1, and the gravitational constant $G = 1$. Define two dimensionless ratios:

$$\pi_1 = \frac{m_1}{m_1 + m_2} \qquad \pi_2 = \frac{m_2}{m_1 + m_2}$$

Thus, we have

$$\pi_1 + \pi_2 = 1$$

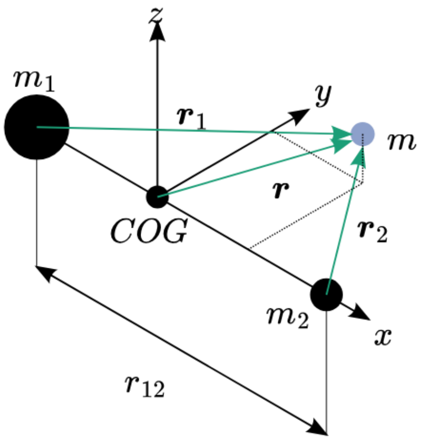


Figure 2: CR3BP coordinates illustration

3.4 Dynamical System

The non-dimensional JWST dynamics in the CR3BP coordinates is governed by

$$\begin{aligned}\ddot{x} - 2\dot{y} &= x - \frac{\pi_1(x+\pi_2)}{\sigma^3} - \frac{\pi_2(x-\pi_1)}{\psi^3} + u_x,\\ \ddot{y} + 2\dot{x} &= y - \frac{\pi_1 y}{\sigma^3} - \frac{\pi_2 y}{\psi^3} + u_y,\\ \ddot{z} &= -\frac{\pi_1 z}{\sigma^3} - \frac{\pi_2 z}{\psi^3} + u_z,\end{aligned}$$

where σ and ψ are the normalized distances from JWST to the Sun and Earth, respectively:

$$\sigma = \sqrt{(x+\pi_2)^2 + y^2 + z^2}, \qquad \psi = \sqrt{(x-\pi_1)^2 + y^2 + z^2}.$$

By defining the state vector $X = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T = [X_1, X_2, X_3, X_4, X_5, X_6]^T$, we get the non-dimensional CR3BP dynamic equations in the state-space form

$$\begin{aligned}\dot{X}_1 &= X_4\\ \dot{X}_2 &= X_5\\ \dot{X}_3 &= X_6\\ \dot{X}_4 &= -\frac{\pi_1}{\sigma^3}(X_1+\pi_2) - \frac{\pi_2}{\psi^3}(X_1-\pi_1) + 2X_5 + X_1 + u_x\\ \dot{X}_5 &= -\frac{\pi_1}{\sigma^3}X_2 - \frac{\pi_2}{\psi^3}X_2 - 2X_4 + X_2 + u_y\\ \dot{X}_6 &= -\frac{\pi_1}{\sigma^3}X_3 - \frac{\pi_2}{\psi^3}X_3 + u_z\end{aligned} \tag{1}$$

which can be viewed as a standard nonlinear dynamical system with control in vector form

$$\dot{X} = f(X) + g(u) \tag{2}$$

With zero control input $u = 0$, the JWST Orbit type is strongly dependent on its initial conditions. We have simulated an unstable Halo Orbit around Earth-Sun L2 point using the the initial condition

$X_{\text{Halo}}(0)$ obtained from the JPL Horizons system (non-dimensionalized) (2):

$$X_{\text{Halo}}(0) = \begin{bmatrix} 1.003\,494\,911\,451\,642\,7 \\ -0.006\,403\,291\,122\,313\,253\,5 \\ 0.007\,161\,610\,846\,661\,841 \\ -0.007\,802\,784\,290\,524\,786 \\ -0.003\,733\,078\,637\,069\,086\,6 \\ -0.016\,701\,355\,598\,003\,984 \end{bmatrix}$$

In addition, we have also simulated a lower Earth orbit (LEO) of the JWST using the initial condition

$$X_{\text{LEO}}(0) = \begin{bmatrix} 1.000\,205\,272\,126\,932\,3 \\ -3.677\,881\,684\,285\,494\,6 \times 10^{-22} \\ -5.960\,562\,510\,814\,000\,6 \times 10^{-2} \\ -2.198\,524\,397\,434\,676\,5 \times 10^{-11} \\ -1.212\,898\,467\,950\,106\,1 \times 10^{-1} \\ 4.403\,023\,282\,003\,022\,5 \times 10^{-23} \end{bmatrix}$$

Notice that in this system, we have that the normalized mass of the Sun, π_1, is equivalent to the x coordinate of the Earth, r_2 (i.e., $\pi_1 = r_2$). Writing the center of mass equation,

$$\pi_1 r_1 + \pi_2 r_2 = 0$$

We have that $r_1 = r_2 - 1$ by the normalization of $r_{12} = r_1 + r_2 = 1$. Substituting this,

$$\pi_1(r_2 - 1) + \pi_2 r_2 = 0$$

$$(\pi_1 + \pi_2) r_2 - \pi_1 = 0$$

By the normalization of masses,

$$r_2 = \pi_1$$

For Sun-Earth system, π_1 is close to 1, the origin of this coordinate is too close to the Sun, making the state X_1 orders of magnitude greater than the others. This will cause numerical instability later in finding the optimal transfer orbit that bridges the LEO and halo orbit. Thus, we shift the origin to the Earth center, so that X_1 would be the same order of magnitude with other states. The new state vector is

$$\tilde{X} = X - \begin{bmatrix} \pi_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} X_1 - \pi_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \end{bmatrix} = \begin{bmatrix} \tilde{X_1} \\ \tilde{X_2} \\ \tilde{X_3} \\ \tilde{X_4} \\ \tilde{X_5} \\ \tilde{X_6} \end{bmatrix}$$

the new state space dynamic becomes

$$
\begin{aligned}
\dot{\tilde{X}}_1 &= \tilde{X}_4 \\
\dot{\tilde{X}}_2 &= \tilde{X}_5 \\
\dot{\tilde{X}}_3 &= \tilde{X}_6 \\
\dot{\tilde{X}}_4 &= -\frac{\pi_1}{\tilde{\sigma}^3}(\tilde{X}_1 + 1) - \frac{\pi_2}{\tilde{\psi}^3}\tilde{X}_1 + 2\tilde{X}_5 + \tilde{X}_1 + \pi_1 + u_x \\
\dot{\tilde{X}}_5 &= -\frac{\pi_1}{\tilde{\sigma}^3}\tilde{X}_2 - \frac{\pi_2}{\tilde{\psi}^3}\tilde{X}_2 - 2\tilde{X}_4 + \tilde{X}_2 + u_y \\
\dot{\tilde{X}}_6 &= -\frac{\pi_1}{\tilde{\sigma}^3}\tilde{X}_3 - \frac{\pi_2}{\tilde{\psi}^3}\tilde{X}_3 + u_z
\end{aligned}
\tag{3}
$$

where we have the transformed JWST-primaries distances as well.

$$
\tilde{\sigma} = \sqrt{(\tilde{X}_1 + \pi_2)^2 + \tilde{X}_2^2 + \tilde{X}_3^2}, \qquad \tilde{\psi} = \sqrt{(\tilde{X}_1 - \pi_1)^2 + \tilde{X}_2^2 + \tilde{X}_3^2}.
$$

Now, we have a well-defined initial value problem with given initial state to solve. Using numerical integration, we can solve for the states at any given time points and make simulation based on them.

3.5 Optimal Control Problem

In our simulations, we aim to include the transfer orbit that bridges the LEO and the L2 halo orbit. Our goal is to adjust the JWST's control inputs so that its state moves from a point in LEO to a point on the halo orbit, governed by the controlled CR3BP dynamics. Among all admissible control sets, we seek one that satisfies the hard constraint of achieving the orbital transfer, while avoiding excessive control effort and transfer time, and ensuring that the state remains within a bounded region. This can be set as a classical optimal control problem that penalizes a quadratic cost function along the trajectory and at the terminal, which is subject to the controlled CR3BP dynamics:

$$
\begin{aligned}
\min_{u \in \mathcal{U}} \int_0^T & \Big[(\tilde{X}(\tau) - \tilde{X}_0)^\top Q (\tilde{X}(\tau) - \tilde{X}_0) \;+\; (u(\tau) - u_0)^\top R (u(\tau) - u_0) \;+\; \beta \Big] d\tau \\
& s.t. \qquad \dot{\tilde{X}} = f(\tilde{X}) + g(u)
\end{aligned}
\tag{4}
$$

where the state weighting matrices $Q \succeq 0$, input weighting matrices $R \succ 0$, and time penalty $\beta > 0$ are assumed to be constant. In addition, $\tilde{X}_0$, u_0 are the reference state and input, where corresponds to zero cost, respectively. τ is the non-dimensionalized time variable and T is the final simulated time. u for all time τ should belong to the space $\mathcal{U} = L^2([0,T]; \mathbb{R}^3)$, square integrable functions mapping the finite time interval $[0,T]$ to the control space $\mathbb{R}^3$.

4 Simulation

4.1 Modules

4.1.1 Python Control System Library

We built the simulation using the `Python Control Systems Library` (`python-control` or `ct`). This package allows the user to define nonlinear input/output systems which are equipped with robust IVP solver. We use this mainly for numerical integration to generate LEO and halo orbit data.

4.1.2 CasADi

CasADi is an open-source tool for nonlinear optimization, algorithmic differentiation, and solving IVP for dynamic system. With these tools, it facilitates rapid — yet efficient — implementation of different methods for numerical optimal control.

4.2 LEO Orbit

In order to resolve the Halo Orbit, the non-dimensionalized system equations of motion are implemented within a computational framework using `ct`. The system dynamics are implemented via the `ct.NonlinearIOSystem` interface. To generate the trajectory, the initial condition $X_{\text{LEO}}(0)$ is also passed as an argument into the solver. The time interval $[0, T]$ is taken to be a total of 1 year with a temporal resolution of 1000 time steps per year. This is also passed into the numerical solver as a `numpy` array generated by `numpy`'s `np.linspace` function.

The numerical integrator is specified as `RK45`, which is a wrapper for `scipy`'s `scipy.integrate.solve_ivp`. Specifically, `RK45` refers to the Dormand-Prince method. This method computes a fourth order solution for the next time step. It then computes a fifth order solution in order to compute an error in order to adaptively change the time step. The error tolerance is dynamically determined as a sum of the absolute tolerance and the relative tolerance scaled by the current state magnitude: $\text{Tol}_i = \text{atol} + \text{rtol}|X_i|$. `RK45` then uses an adaptive time step derived from such tolerance for the next time step. In particular, a value of 1e-12 was used for atol and a value of 1e-9 The butcher tableau for this method is shown below.

$$
\begin{array}{c|ccccccc}
0 & & & & & & & \\
1/5 & 1/5 & & & & & & \\
3/10 & 3/40 & 9/40 & & & & & \\
4/5 & 44/45 & -56/15 & 32/9 & & & & \\
8/9 & 19372/6561 & -25360/2187 & 64448/6561 & -212/729 & & & \\
1 & 9017/3168 & -355/33 & 46732/5247 & 49/176 & -5103/18656 & & \\
1 & 35/384 & 0 & 500/1113 & 125/192 & -2187/6784 & 11/84 & \\
\hline
X_i^{(5)} & 35/384 & 0 & 500/1113 & 125/192 & -2187/6784 & 11/84 & 0 \\
X_i^{(4)} & 5179/57600 & 0 & 7571/16695 & 393/640 & -92097/339200 & 187/2100 & 1/40
\end{array}
\tag{5}
$$

4.3 Transfer Orbit

For solving the challenging transfer orbit problem described by Sec. 3.5, we reformulate the optimal control problem (OCP) as a nonlinear programming (NLP) problem using `CasADi`. In brief, `CasADi` discretizes the system dynamics in time, treating the Horizon T, the state X and control input u at each time node as independent decision variables. A numerical integrator is then used to propagate the dynamics from one time node to the next, and continuity of the state trajectory is enforced by imposing hard equality constraints that match the terminal state of the propagation with the state variable at the subsequent time node.

Throughout the horizon, the cost function is computed from the states and control inputs at all time steps, and the built-in automatic differentiation of CasADi is used to compute the cost gradient

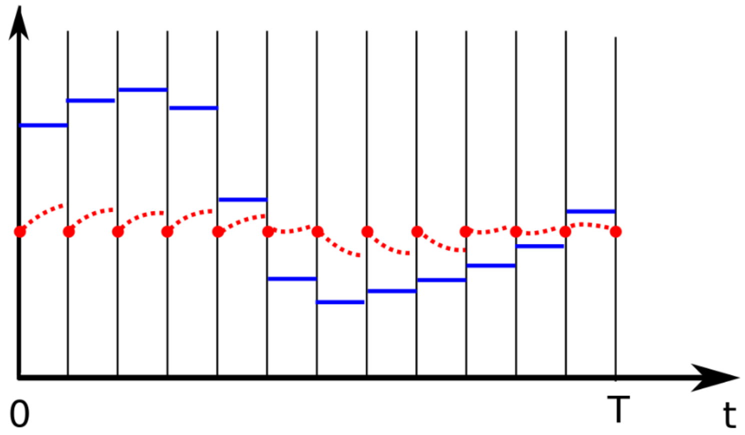


Figure 3: In direct multiple shooting method, we discretize the dynamic, and treat the state and input as free parameters to be optimized. At each time step, the states are propagated by a numerical integrator. The previous end state and next starting state are set to be equality constraints.

with respect to all decision variables. These gradients are then supplied to a standard NLP solver (e.g., IPOPT). We can retrieve the optimal trajectory X^*, optimal control u^*, and Optimal Horizon T^*. This procedure is called the multiple shooting method, a widely used numerical approach for solving optimal control problems.

4.4 Halo Orbit

The Halo orbit is performed in a similar manner as the LEO orbit, with the exception that the initial condition $X_{\text{Halo}}(0)$ is used instead and 10 years are simulated.

4.4.1 Matplotlib + PyVista

Finally, for orbit visualization, we use Matplotlib to efficiently plot the trajectory data, which allows for the more effective inspection and validation of the results of orbit optimization. We overlay critical markers such as the Earth, the Sun and the L_2 point. The class `ThreeBodySystem` implements a function `RotToFixed` that performs coordinate transformations from the rotating frame into the fixed frame and animations of both are plotted. For spatial presentation, we employ PyVista to generate high-fidelity 3D animations. This allows us to map high-resolution textures onto the celestial bodies and visually trace the Halo orbit over time, offering a clear qualitative perspective on the simulation.

5 Experimental results and validation

5.1 Validation of the Simulator

Before running any of our experiments, we carried out a series of validation checks to assess whether our implementation behaves consistently with known properties of the Sun–Earth CR3BP. The goal of these checks is not to prove the simulator correct in an absolute sense, but to establish credibility and increase confidence that the numerical model reproduces the expected dynamics. Our validation follows the categories discussed in lecture, including face validation, behavior validation, and analytical-model validation.

1. Face Validation: LEO-to-Halo Trajectory Shape. A first qualitative check comes from visually inspecting the simulated LEO-to-halo trajectory. In the rotating frame, halo orbits are known to form a tilted, three-dimensional, highly eccentric loop around L_2. The trajectory generated by our simulator exhibits exactly this structure: the transfer arc departs from a nearly circular LEO, transitions outward along an unstable manifold-like path, and settles into the characteristic halo-orbit geometry. The overall shape, inclination, and amplitude match reference CR3BP halo trajectories published in the literature. This visual agreement provides a strong form of *face validation*, indicating that the simulated motion behaves as expected to an expert observer.

2. Jacobi Constant Preservation (Analytical-model validation). The Jacobi constant C is an invariant of the CR3BP for pure gravitational motion in the rotating frame. This provides a direct analytical benchmark for validating our numerical integrator and dynamical equations. We compute $C(t)$ along the simulated trajectory and measure the drift $C(t) - C(0)$ over a two-year window.

As shown in Figure 4, the relative deviation remains on the order of 10^{-12}, which is consistent with floating-point roundoff for a 4th–5th order Runge–Kutta scheme. This test strongly supports that our implementation of the CR3BP equations and nondimensional scaling is correct and numerically stable.

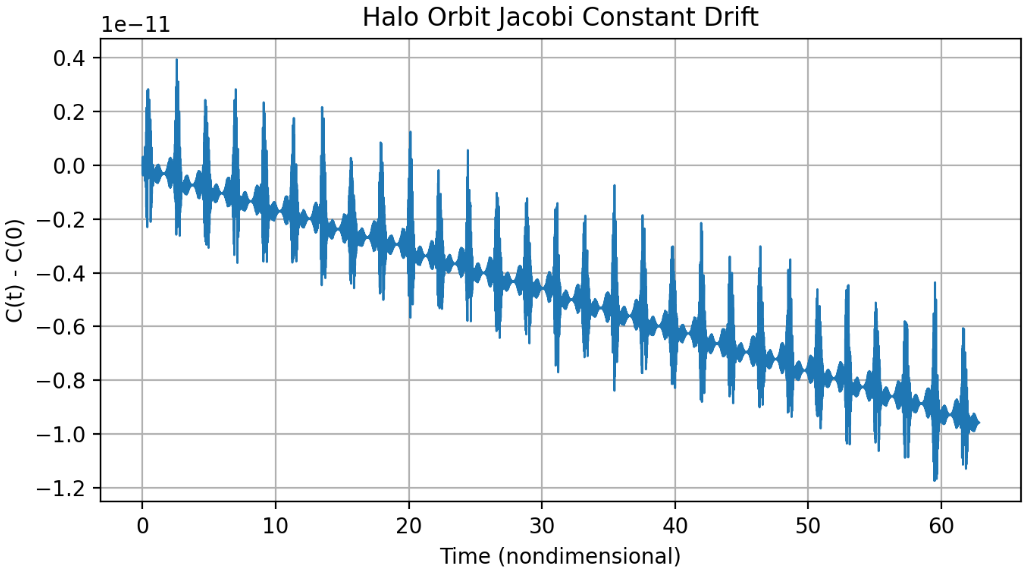


Figure 4: Jacobi constant drift $C(t) - C(0)$ over the simulated halo trajectory. Deviations remain on the order of 10^{-12}, consistent with numerical roundoff.

3. Halo-Orbit Boundedness (Behavior validation). A physically accurate halo orbit must remain bounded around the L_2 point without drifting away. This provides a behavior-based validation test: if the trajectory remains within a stable range and oscillates periodically, the simulator is capturing the correct qualitative dynamics.

We compute the instantaneous distance

$$r(t) = \|X(t) - X_{L_2}\|$$

and compare it to the initial distance r_0. As shown in Figure 5, the radius oscillates periodically between approximately $0.80r_0$ and $1.00r_0$ over multiple Halo cycles. This is consistent with the amplitude behavior of large Sun–Earth halo orbits. The absence of any long-term growth or decay indicates that the dynamics remain bounded, providing a strong *behavior validation* signal.

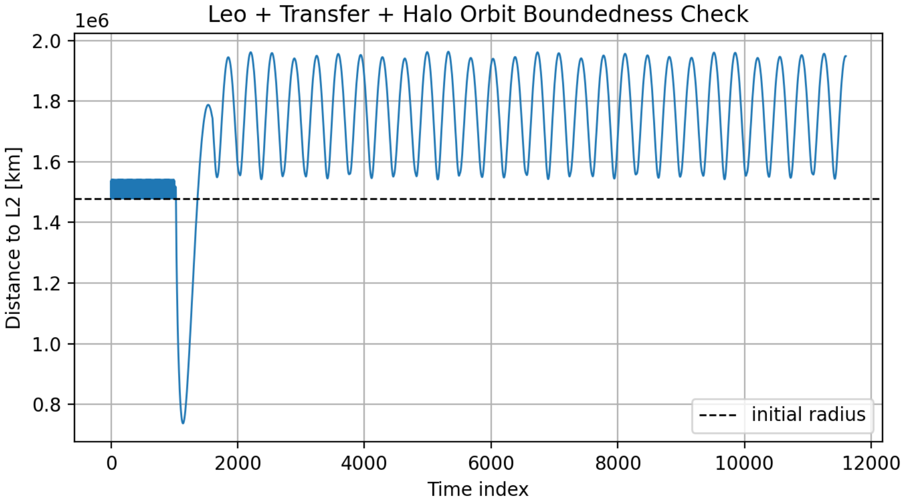
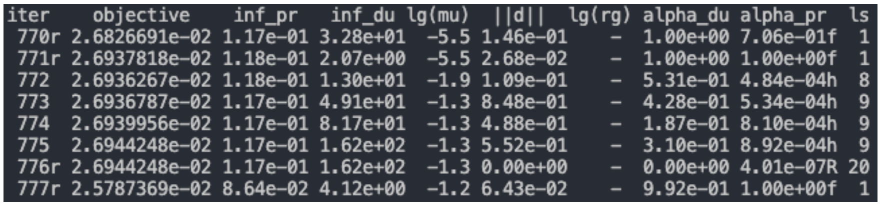


Figure 5: Distance from the spacecraft to L_2 over the simulated leo, transfer, and halo orbit. The motion remains bounded and periodic, with no evidence of drift away from the equilibrium region.

Together, these validation steps—visual inspection of trajectory geometry, analytical preservation of the Jacobi invariant, and boundedness of the halo orbit—provide complementary evidence that our simulator is producing credible CR3BP dynamics. These checks align with the verification and validation guidelines discussed in lecture and establish a reliable foundation for the transfer-orbit and LEO-to-L_2 studies that follow.

4. Optimization To compute the transfer orbit, we reformulate the original OCP as a nonlinear programming (NLP) problem and solve it using a CasADi plugin IPOPT. IPOPT iteratively searches for a local optimum using the gradient information of the cost function, and gradually approaching a point that satisfies the Karush–Kuhn–Tucker (KKT) conditions. At each iteration, the solver prints a log containing the current objective value, primal infeasibility, dual infeasibility, the step norm $\|d\|$. The primal infeasibility measures the degree to which the current iterate violates the constraints, while the dual infeasibility quantifies the violation of the stationarity (optimality) conditions. When IPOPT converges to a feasible locally optimal solution, both the primal and dual infeasibilities are expected to fall below their prescribed tolerances.

In a representative optimization run, we set the cost function to be the form of (4) except using sum instead of integral, and letting $Q = diag([10, 10, 10, 1, 1, 1])$, $R = diag([1, 1, 1])$ and $\beta = 20$.



```
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
 770r 2.6826691e-02 1.17e-01 3.28e+01  -5.5 1.46e-01    -  1.00e+00 7.06e-01f  1
 771r 2.6937818e-02 1.18e-01 2.07e+00  -5.5 2.68e-02    -  1.00e+00 1.00e+00f  1
 772  2.6936267e-02 1.18e-01 1.30e+01  -1.9 1.09e-01    -  5.31e-01 4.84e-04h  8
 773  2.6936787e-02 1.17e-01 4.91e+01  -1.3 8.48e-01    -  4.28e-01 5.34e-04h  9
 774  2.6939956e-02 1.17e-01 8.17e+01  -1.3 4.88e-01    -  1.87e-01 8.10e-04h  9
 775  2.6944248e-02 1.17e-01 1.62e+02  -1.3 5.52e-01    -  3.10e-01 8.92e-04h  9
 776r 2.6944248e-02 1.17e-01 1.62e+02  -1.3 0.00e+00    -  0.00e+00 4.01e-07R 20
 777r 2.5787369e-02 8.64e-02 4.12e+00  -1.2 6.43e-02    -  9.92e-01 1.00e+00f  1
```

Figure 6: IPOPT log showing convergence to a nearly local optimal solution

The IPOPT log is shown in the figure below, we observe that the primal and dual infeasibilities decrease to sufficiently small values at iteration 777, which corresponds to a nearly local optimal.

As shown in Fig.7, the small circular orbit around the blue dot (Earth) is the LEO. The large circle is the halo orbit around L2 of Sun-Earth system. The curve that starts from LEO and is smoothly injected into the side of the halo orbit is the optimized transfer orbit. This transfer orbit is smooth, and does not deviate too far away from its destination. The transfer orbit's time of flight, T^*, is 0.73 years, which is reasonable.

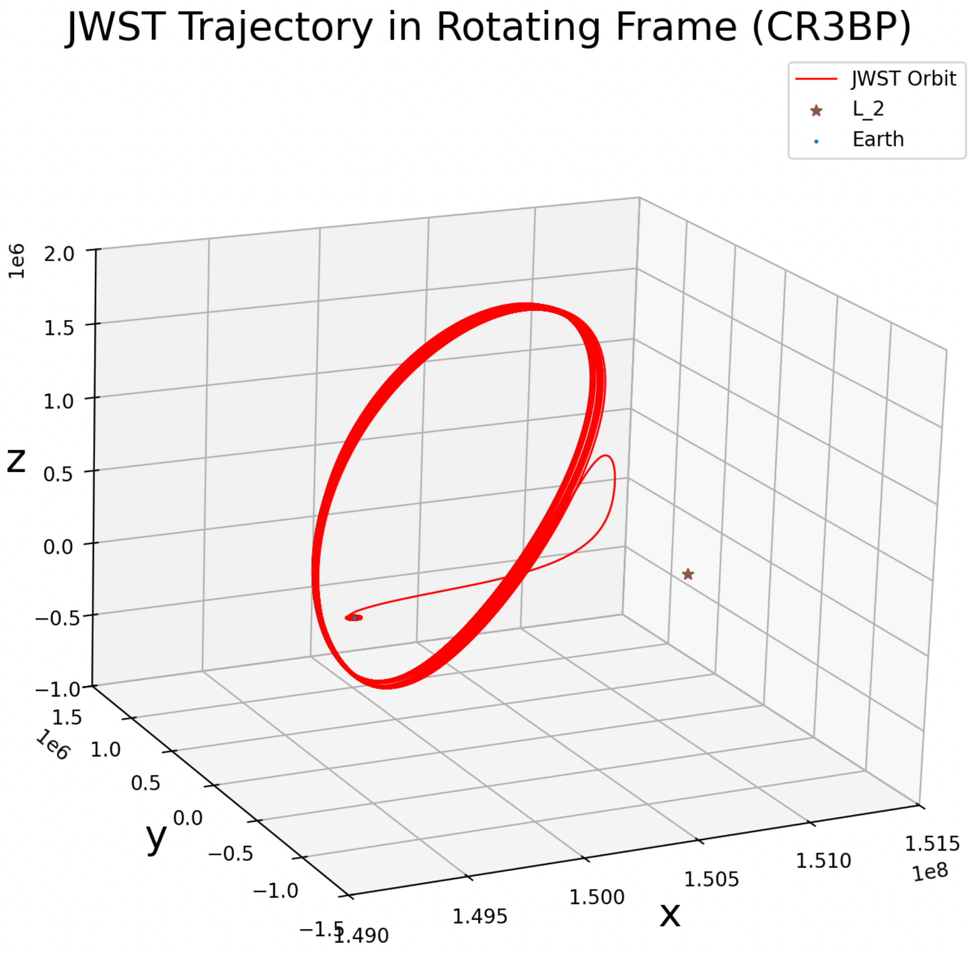


Figure 7: Optimized transfer orbit bridges the LEO and halo orbit in rotating frame

As shown in Fig.8, the control input saturates at beginning because the spacecraft requires energy to raise its orbit. Since, in real life, spacecraft cannot infinitely accelerates, we set the bounds for u_x, u_y, and u_z to be in $[-0.1, 0.1]$ in the optimization problem.

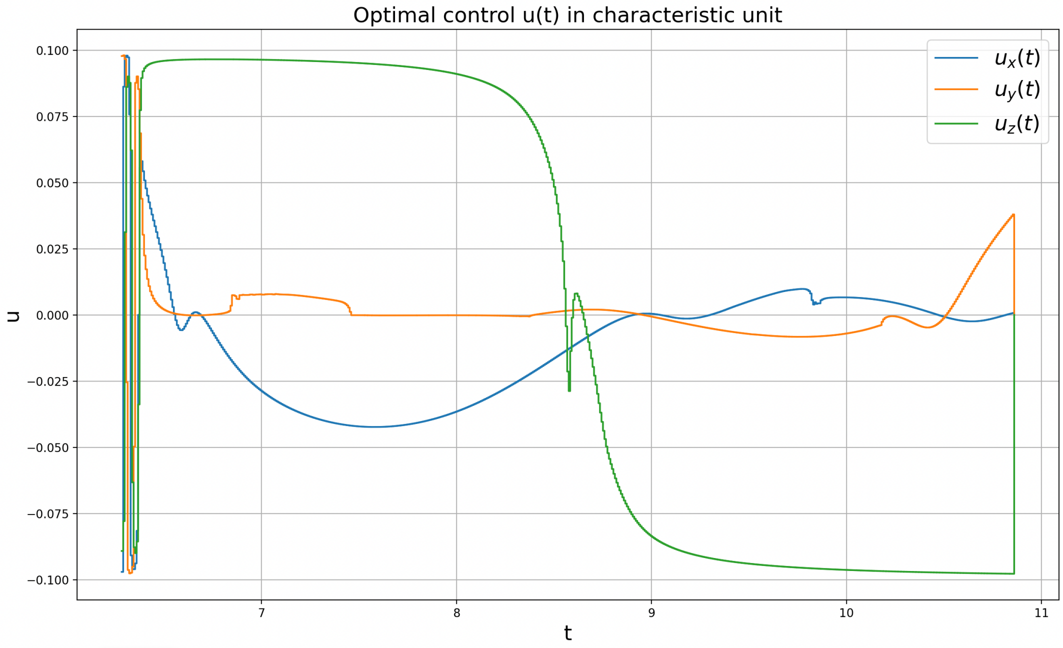


Figure 8: Optimal control during transfer orbit in characteristic unit

6 Discussion, conclusions, summary

6.1 Conclusions

6.1.1 Lagrange point

From this simulation, we learn that in the CR3BP framework, even though the L2 point is unstable, there exist stable limit cycles in its vicinity, among which the halo orbit is a prominent example. Such trajectories play a critical role in specific mission scenarios.

6.1.2 Optimization

We also observe that when using IPOPT to solve optimal control problems, the solver requires a user-provided initial guess, and finding a reasonable initial guess can be nearly as challenging as solving the optimization problem itself. This motivates a continuation or homotopy-style strategy, where we solve a sequence of related problems and use the result from the previous optimization as the initial guess for the next.

In addition, variable scaling is crucial for successful optimization. Poor scaling can induce numerical instability in the IVP solver and lead to noisy gradients from algorithmic differentiation, which in turn can hinder the identification of a valid descent direction or even cause the optimization solver to abort. Identifying appropriate scalings or performing suitable coordinate transformations is the first step in formulating a well-posed optimization problem. This requires user to have an insight of the problem. Finally, the constraints setting and the choice of cost function are also essential for achieving robust convergence of the optimization.

6.2 Limitation & Future Work

In this simulation, all celestial mechanics–related parameters are drawn from publicly available sources such as NASA JPL Horizon system(2). By contrast, the spacecraft properties themselves are fictitious. To apply this framework to real mission design, both the cost function and the variable bounds in the optimization are related to spacecraft itself, we need to consider the limitation of the spacecraft. The optimization would need to be tailored to the actual spacecraft characteristics—for example, its mass, propulsion system, and available propellant.

Moreover, the CR3BP model is only an approximate description of the dynamics. In practice, one must also account for perturbations from other bodies in the solar system, the J2 effect of Planets, the relativistic effects, and other higher-order influences to obtain a more accurate dynamical model.

Finally, computer simulations inevitably introduce numerical errors. Thus, for a real mission, the spacecraft need to use a combination of feedforward control, feedback control, and real-time measurement data to robustly track the designed transfer orbit.

References

[1] David Folta, Steven Cooley, Kathleen Howell, and Frank H. Bauer. Trajectory design strategies for the NGST l2 libration point mission. In *AAS/AIAA Spaceflight Mechanics Meeting*, Santa Barbara, CA, USA, 2001. AAS Paper 01-206. URL: https://ntrs.nasa.gov/api/citations/20010026414/downloads/20010026414.pdf.

[2] J. D. Giorgini. Three-body periodic orbits (jpl horizons tool). https://ssd.jpl.nasa.gov/tools/periodic_orbits.html#/periodic. Accessed: 2025-10-26.

[3] Kathleen C. Howell, David L. Mains, and Brian T. Barden. Transfer trajectories from earth parking orbits to sun–earth halo orbits. In *AAS/AIAA Spaceflight Mechanics Meeting*, Cocoa Beach, FL, USA, 1994. AAS Paper 94-160, accessed 2025-11-23. URL: https://engineering.purdue.edu/people/kathleen.howell.1/Publications/Conferences/1994_AAS_HowMaiBar.pdf.

[4] NASA. Webb's orbit at sun–earth lagrange point 2 (l2) — illustration. https://science.nasa.gov/asset/webb/webbs-orbit-at-sun-earth-lagrange-point-2-l2/, 2021. Accessed: 2025-11-23.

[5] NASA. Webb's orbit. https://science.nasa.gov/mission/webb/orbit/, 2025. Page last updated Apr. 30, 2025; accessed 2025-11-23.

[6] P. Nath and R. V. Ramanan. Precise halo orbit design and optimal transfer to halo orbits from earth using differential evolution. *Advances in Space Research*, 57(1):202–217, 2016. URL: https://www.researchgate.net/publication/283477421_Precise_Halo_orbit_Design_and_Optimal_Transfer_to_Halo_Orbits_from_Earth_Using_Differential_Evolution.

[7] John E. Prussing and Bruce A. Conway. *Orbital Mechanics*. Oxford University Press, New York, NY, 1993. URL: https://global.oup.com/ushe/product/orbital-mechanics-9780197788974.

[8] python-control developers. Python control systems library. https://python-control.readthedocs.io/en/0.10.2/, 2024. Version 0.10.2, accessed 2025-11-23.

[9] PyVista Developers. Pyvista example: Orbiting. https://docs.pyvista.org/examples/02-plot/orbit, 2017. Accessed: 2025-11-23.

[10] Raoul R. Rausch. Earth to halo orbit transfer trajectories. Master's thesis, Purdue University, West Lafayette, IN, USA, 2005. Accessed: 2025-11-23. URL: https://engineering.purdue.edu/people/kathleen.howell.1/Publications/Masters/2005_Rausch.pdf.

[11] Ben Weber. Circular restricted three-body problem. https://orbital-mechanics.space/the-n-body-problem/circular-restricted-three-body-problem.html. Accessed: 2025-10-23.

Appendix: Division of Work

Our team divided the work according to each member's strengths, while ensuring everyone contributed to both technical tasks and writing. Tianyang Hu was responsible for the core system implementation, including setting up the CR3BP simulator and running the main experiments. Haoran handled system validation, designing and performing the numerical checks for the transfer and halo orbit. Mark focused on the 3D trajectory visualization, building the PyVista-based plots and animations. Shiqi designed and implemented the video gallery user interface, and produced the final summary video. Although each person contributed in different functional areas, all team members contributed a similar amount of effort to the project.