SGN – Assignment #1

Marcello Pareschi, 252142

1 Periodic orbit

Exercise 1

Consider the 3D Earth–Moon Circular Restricted Three-Body Problem with $\mu = 0.012150$. Note that the CRTBP has an integral of motion, that is, the Jacobi constant

$$J(x, y, z, v_x, x_y, v_z) := 2\Omega(x, y, z) - v^2 = C$$

where $\Omega(x, y, z) = \frac{1}{2}(x^2 + y^2) + \frac{1-\mu}{r_1} + \frac{\mu}{r_2} + \frac{1}{2}\mu(1-\mu)$ and $v^2 = v_x^2 + v_y^2 + v_z^2$.

1) Find the coordinates of the five Lagrange points L_i in the rotating, adimensional reference frame with at least 10-digit accuracy and report their Jacobi constant C_i.

Solutions to the 3D CRTBP satisfy the symmetry

$$\mathcal{S} : (x, y, z, \dot{x}, \dot{y}, \dot{z}, t) \to (x, -y, z, -\dot{x}, \dot{y}, -\dot{z}, -t).$$

Thus, a trajectory that crosses perpendicularly the $y = 0$ plane twice is a periodic orbit.

2) Given the initial guess $\mathbf{x}_0 = (x_0, y_0, z_0, v_{x0}, v_{y0}, v_{z0})$, with

$$\begin{aligned} x_0 &= 1.068792441776 \\ y_0 &= 0 \\ z_0 &= 0.071093328515 \\ v_{x0} &= 0 \\ v_{y0} &= 0.319422926485 \\ v_{z0} &= 0 \end{aligned}$$

Find the periodic halo orbit having a Jacobi Constant $C = 3.09$; that is, develop the theoretical framework and implement a differential correction scheme that uses the STM, either approximated through finite differences **or** achieved by integrating the variational equation.
Hint: Consider working on $\varphi(\mathbf{x} + \Delta\mathbf{x}, t + \Delta t)$ and $J(\mathbf{x} + \Delta\mathbf{x})$ and then enforce perpendicular cross of $y = 0$ and Jacobi energy.

The periodic orbits in the CRTBP exist in families. These can be computed by 'continuing' the orbits along one coordinate or one parameter, e.g., the Jacobi energy C. The *numerical continuation* is an iterative process in which the desired variable is *gradually* varied, while the rest of the initial guess is taken from the solution of the previous iteration, thus aiding the convergence process.

3) By gradually decreasing C and using numerical continuation, compute the families of halo orbits until $C = 3.04$.
(8 points)

1.1 Lagrange Points Location

The coordinates of the five Lagrange points can be computed by finding the stationary points of the scalar potential function $\Omega(x, y, z)$, described in Appendix A. These points correspond to the zeros of the gradient of Ω. Notably, it can be shown that $\dfrac{\partial \Omega}{\partial z} = 0$ if and only if $z = 0$, reducing the problem to finding the stationary points of Ω in the x-y plane. Defining the vector field:

$$\mathbf{U}(x,y) = \begin{bmatrix} \frac{\partial \Omega}{\partial x}\Big|_{z=0} \\ \frac{\partial \Omega}{\partial y}\Big|_{z=0} \end{bmatrix} = \begin{bmatrix} x - \dfrac{(1-\mu)(x+\mu)}{[(x+\mu)^2+y^2]^{\frac{3}{2}}} - \dfrac{\mu(x+\mu-1)}{[(x+\mu-1)^2+y^2]^{\frac{3}{2}}} \\ y - \dfrac{(1-\mu)y}{[(x+\mu)^2+y^2]^{\frac{3}{2}}} - \dfrac{\mu y}{[(x+\mu-1)^2+y^2]^{\frac{3}{2}}} \end{bmatrix}, \tag{1}$$

the problem consists in finding the zeros of $\mathbf{U}(x, y)$. To solve this zero-finding problem, the MATLAB function `fsolve` was employed. To ensure convergence with at least 10-digit accuracy, the `OptimalityTolerance` option of `fsolve` was set to 10^{-10}. The initial guesses for this root-finding problem, reported in Table 1, were chosen by inspecting the contour plot of U, the norm of the vector field defined in Equation 1. As shown in Figure 1, $\mathbf{U}(x, y)$ has five distinct zeros, corresponding to the Lagrange points. The Jacobi constant for each Lagrange point was computed using the determined coordinates while setting all the velocity components to zero. The computed coordinates and corresponding Jacobi constants are summarized in Table 2.

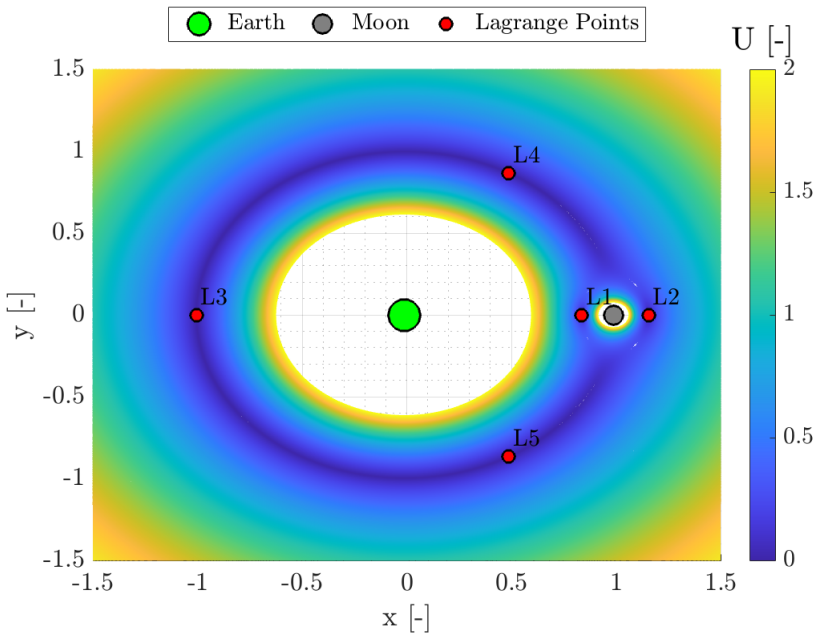


Figure 1: Lagrange points in the Earth-Moon rotating frame and U contour plot

Table 1: Initial guesses coordinates in the Earth-Moon rotating frame

$L_i^{(0)}$	x	y
$L_1^{(0)}$	0.0	0.0
$L_2^{(0)}$	1.5	0.0
$L_3^{(0)}$	-1.5	0.0
$L_4^{(0)}$	0.0	1.5
$L_5^{(0)}$	0.0	-1.5

Table 2: Lagrangian points coordinates in the Earth-Moon rotating frame and Jacobi constants

	L_1	L_2	L_3	L_4	L_5
x	0.8369180073	1.1556799131	−1.0050624018	0.48785000000	0.48785000000
y	0.0000000000	0.0000000000	0.0000000000	0.8660254038	−0.8660254038
C	3.2003380950	3.1841582164	3.0241489429	3.0000000000	3.0000000000

1.2 Halo Orbit Computation

To determine the correct initial state that leads to a periodic halo orbit with a Jacobi constant of $C = 3.09$, a differential correction was implemented. This correction adjusts the values of x_0, z_0, and v_{y0} while keeping y_0, v_{x0}, and v_{z0} equal to zero. This approach is based on the condition, stated in the exercise, that a periodic orbit must cross the x-z plane perpendicularly, i.e., with velocity parallel to the y axis, twice.
The differential correction scheme was developed by linearizing the flow of the ODE governing the Circular Restricted Three-Body Problem dynamics (Appendix B) and the Jacobi constant expression (Appendix C) with respect to the initial condition $\mathbf{x}_0$ and the propagation time t_f:

$$\begin{cases} \mathbf{x}(t) = \varphi(\mathbf{x}_0 + \delta\mathbf{x}_0, t_0, t_f + \delta t_f) \approx \varphi(\mathbf{x}_0, t_0, t_f) + \dfrac{\partial\varphi(\mathbf{x}_0, t_0, t_f)}{\partial\mathbf{x}_0}\delta\mathbf{x}_0 + \dfrac{\partial\varphi(\mathbf{x}_0, t_0, t_f)}{\partial t_f}\delta t_f \\ C(\mathbf{x}_0 + \delta\mathbf{x}_0) \approx C(\mathbf{x}_0) + \dfrac{\partial C(\mathbf{x}_0)}{\partial\mathbf{x}_0}\delta\mathbf{x}_0 + \dfrac{\partial C(\mathbf{x}_0)}{\partial t_f}\delta t_f \end{cases} \tag{2}$$

From Equation 2, the linearized equations describing the displacements $\delta\mathbf{x}_f$ and δC from the nominal condition can be expressed in terms of the displacements $\delta\mathbf{x}_0$ and δt_f from the nominal conditions of $\mathbf{x}_0$ and t_f, respectively:

$$\begin{bmatrix} \delta\mathbf{x}_f \\ \delta C \end{bmatrix} = \begin{bmatrix} \Phi(0, t_f) & \dot{\mathbf{x}}_f \\ \dfrac{\partial J}{\partial\mathbf{x}_0} & 0 \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}_0 \\ \delta t_f \end{bmatrix} \tag{3}$$

Finally, since the correction is applied only to x_0, z_0, v_{y0}, and t_f, and the conditions for the periodicity of the orbit and for reaching the target Jacobi constant are imposed on y_f, v_{xf}, v_{zf}, and C, it is possible to extract a subset of equations from Equation 3. This leads to the linear system reported in Equation 4, which is repeatedly solved in the iterative process described in Algorithm 1, necessary because Equation 3 comes from a linearization of the dynamics, meaning that it cannot converge in a single step:

$$\begin{bmatrix} \Phi_{21} & \Phi_{23} & \Phi_{25} & \dot{y}_f \\ \Phi_{41} & \Phi_{43} & \Phi_{45} & \dot{v}_{xf} \\ \Phi_{61} & \Phi_{63} & \Phi_{65} & \dot{v}_{zf} \\ \mathbf{J}'(1) & \mathbf{J}'(3) & \mathbf{J}'(5) & 0 \end{bmatrix} \begin{bmatrix} \delta x_0 \\ \delta z_0 \\ \delta v_{y0} \\ \delta t_f \end{bmatrix} = \begin{bmatrix} \varepsilon_{yf} \\ \varepsilon_{v_{xf}} \\ \varepsilon_{v_{zf}} \\ \varepsilon_C \end{bmatrix} \tag{4}$$

Algorithm 1 Find Halo Orbit Initial Conditions

Require: First guess $\mathbf{x}_0^{(0)}$, $t_f^{(0)}$, C_{target}
Ensure: Adjusted state $\mathbf{x}_0$, final time t_f, iterations number k

1: Initialize iter $= 0$, errors $\varepsilon_y^{(0)}$, $\varepsilon_y^{(0)}$, $\varepsilon_{v_{xf}}^{(0)}$, $\varepsilon_{v_{zf}}^{(0)}$, $\varepsilon_C^{(0)} = 1$, and tolerance tol $= 10^{-13}$
2: **while** $\max\left(\left|\varepsilon_y^{(k)}\right|, \left|\varepsilon_{v_{xf}}^{(k)}\right|, \left|\varepsilon_{v_{zf}}^{(k)}\right|, \left|\varepsilon_C^{(k)}\right|\right) > tol$ **and** iter $< N_{max}$ **do**
3: Update $\mathbf{x}_0^{(k)}$ and $t_f^{(k)}$ with corrections $\delta x_0^{(k-1)}$, $\delta z_0^{(k-1)}$, $\delta v_{y0}^{(k-1)}$, $\delta t_f^{(k-1)}$
4: Propagate $\mathbf{x}_0^{(k)}$ up to $t_f^{(k)}$, compute $\Phi(0, t_f^{(k)})$ and $\mathbf{J}'(\mathbf{x}_0^{(k)})$
5: Compute errors: $\varepsilon_{yf}^{(k)}$, $\varepsilon_{v_{xf}}^{(k)}$, $\varepsilon_{v_{zf}}^{(k)}$, $\varepsilon_C^{(k)}$
6: Solve Equation 4 and find $\delta x_0^{(k)}$, $\delta z_0^{(k)}$, $\delta v_{y0}^{(k)}$, $\delta t_f^{(k)}$
7: $k = k + 1$
8: **end while**

In Equation 4, Φ_{ij} are the i-j components of the State Transition Matrix $\Phi(0, t_f)$ and $\mathbf{J}'(i)$ are

the i-th components of the Jacobi constant Jacobian. The computation of Φ and $\mathbf{J}'$ at each step of Algorithm 1 was performed numerically using a forward finite differences scheme.

It's important to note that $t_0^{(0)}$ was not provided by the text of exercise, thus it was computed by propagating $\mathbf{x}_0^{(0)}$ up to the x-z plane crossing, stopping the integration with an event function.

At the (k)-th step of the `while` loop in Algorithm 1, the initial condition $\mathbf{x}_0^{(k)}$ is propagated up to $t_f^{(k)}$ by numerical integration of the CRTBP equations. Subsequently, Equation 4 is solved to determine the corrections required for the initial condition and propagation time to bridge the gap between the final state of the propagation and the target state. At the $(k+1)$-th step of the algorithm, the initial condition and propagation time are updated using the corrections computed in the previous step. This iterative process continues until convergence is achieved.

The resulting orbit, which initial states are reported in Table 3 in the rotating, adimensionalized reference frame is depicted in Figure 2.

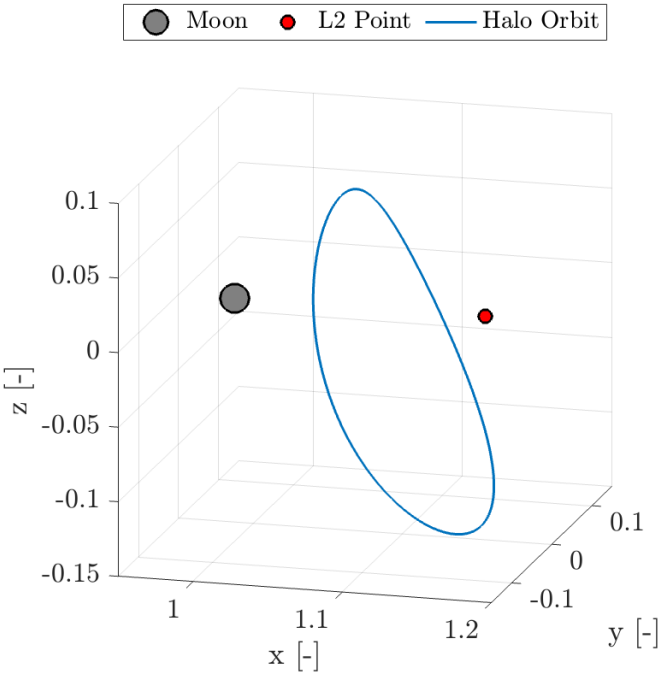


Figure 2: Halo orbit with $C = 3.09$ in the Earth-Moon rotating frame

Table 3: Corrected initial state of the halo orbit with $C = 3.09$ in the Earth-Moon rotating frame

x_0	1.0590402077
y_0	0.0000000000
z_0	0.0739277378
v_{x0}	0.0000000000
v_{y0}	0.3469245709
v_{z0}	0.0000000000

1.3 Numerical Continuation

To determine the initial condition corresponding to the halo orbit with a Jacobi constant of $C = 3.04$ while ensuring fast convergence of the algorithm, the numerical continuation technique was employed. Starting from the initial guess found in subsection 1.2, Algorithm 1 was iteratively executed, with the target Jacobi constant gradually decreased from 3.09 to 3.04. The initial condition obtained at each step of this process was used as the starting guess for the subsequent step. As a result, a family of six halo orbits with Jacobi constants equally spaced between 3.09 and 3.04 was identified. The resulting orbits are depicted in the rotating, adimensionalized reference frame in Figure 3. The initial states corresponding to the halo orbit with $C = 3.04$ are reported in Table 4.

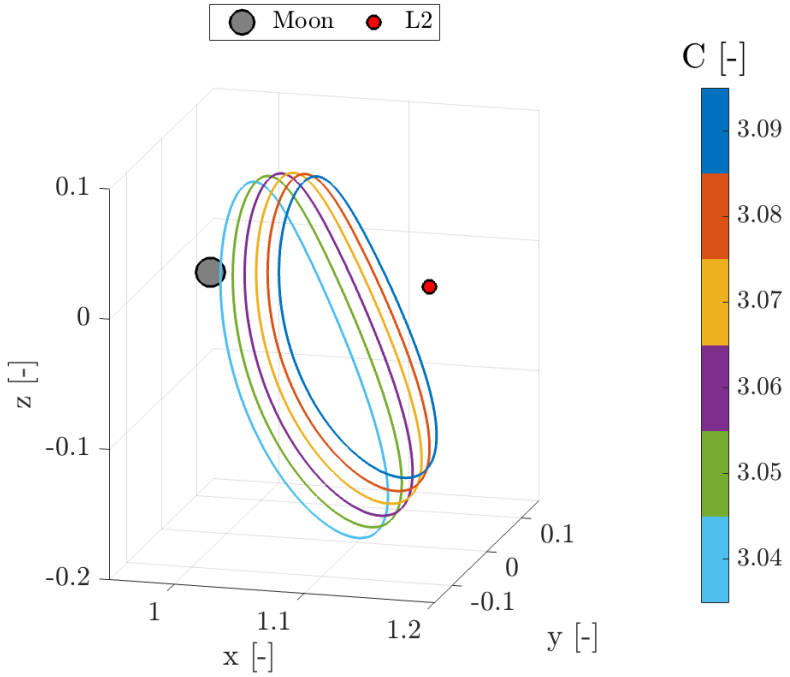


Figure 3: Family of halo orbits in the Earth-Moon rotating frame

Table 4: Corrected initial state of the halo orbit with $C = 3.04$ in the Earth-Moon rotating frame

x_0	1.0125655235
y_0	0.0000000000
z_0	0.0672339583
v_{x0}	0.0000000000
v_{y0}	0.5103251959
v_{z0}	0.0000000000

2 Impulsive guidance

Exercise 2

Consider the two-impulse transfer problem stated in Section 3.1 (Topputo, 2013)*.

1) Using the procedure in Section 3.2, produce a first guess solution using $\alpha = 0.2\pi$, $\beta = 1.41$, $\delta = 4$, and $t_i = 2$. Plot the solution in both the rotating frame and Earth-centered inertial frame (see Appendix 1 in (Topputo, 2013)). Consider the parameters listed in Table 5 and extract the radius and gravitational parameters of the Earth and Moon from the provided kernels and use the latter to compute the parameter μ.

Symbol	Value	Units	Meaning
m_s	3.28900541×10^5	-	Scaled mass of the Sun
ρ	3.88811143×10^2	-	Scaled Sun-(Earth+Moon) distance
ω_s	$-9.25195985 \times 10^{-1}$	-	Scaled angular velocity of the Sun
ω_{em}	$2.66186135 \times 10^{-1}$	s^{-1}	Earth-Moon angular velocity
l_{em}	3.84405×10^8	m	Earth-Moon distance
h_i	167	km	Altitude of departure orbit
h_f	100	km	Altitude of arrival orbit
DU	3.84405000×10^5	km	Distance Unit
TU	4.34256461	days	Time Unit
VU	1.02454018	km/s	Velocity Unit

Table 5: Constants to be considered to solve the PBRFBP. The units of distance, time, and velocity are used to map scaled quantities into physical units.

2) Considering the first guess in 1) and using $\{\mathbf{x}_i, t_i, t_f\}$ as variables, solve the problem in Section 3.1 with simple shooting in the following cases
   a) without providing any derivative to the solver, and
   b) by providing the derivatives and by estimating the state transition matrix with variational equations.
3) Considering the first guess solution in 1) and the procedure in Section 3.3, solve the problem with multiple shooting taking $N = 4$ and using the variational equation to compute the Jacobian of the nonlinear equality constraints.
4) Perform an n-body propagation using the solution $\{\mathbf{x}_i, t_i, t_f\}$ obtained in point 2), transformed in Earth-centered inertial frame and into physical units. To move from 2-D to 3-D, assume that the position and velocity components in inertial frame are $r_z(t_i) = 0$ and $v_z(t_i) = 0$. To perform the propagation it is necessary to identify the epoch t_i. This can be done by mapping the relative position of the Earth, Moon and Sun in the PCRTBP to a similar condition in the real world:
   a) Consider the definition of $\theta(t)$ provided in Section 2.2 to compute the angle $\theta_i = \theta(t_i)$. Note that this angle corresponds to the angle between the rotating frame x-axis, aligned to the position vector from the Earth-Moon System Barycenter (EMB) to the Moon, and the Sun direction.
   b) The angle θ ranges between $[0, 2\pi]$ and it covers this domain in approximately the revolution period of the Moon around the Earth.

*F. Topputo, "On optimal two-impulse Earth–Moon transfers in a four-body model", Celestial Mechanics and Dynamical Astronomy, Vol. 117, pp. 279–313, 2013, DOI: 10.1007/s10569-013-9513-8.

c) Solve a zero-finding problem to determine the epoch at which the angle Moon-EMB-Sun is equal to θ_i, considering as starting epoch `2024 Sep 28 00:00:00.000 TDB`. **Hints**: Exploit the SPK kernels to define the orientation of the rotating frame axes in the inertial frame for an epoch t. Consider only the projection of the EMB-Sun position vector onto the so-defined x-y plane to compute the angle (planar motion).

Plot the propagated orbit and compare it to the previously found solutions. (11 points)

2.1 Initial Guess Generation

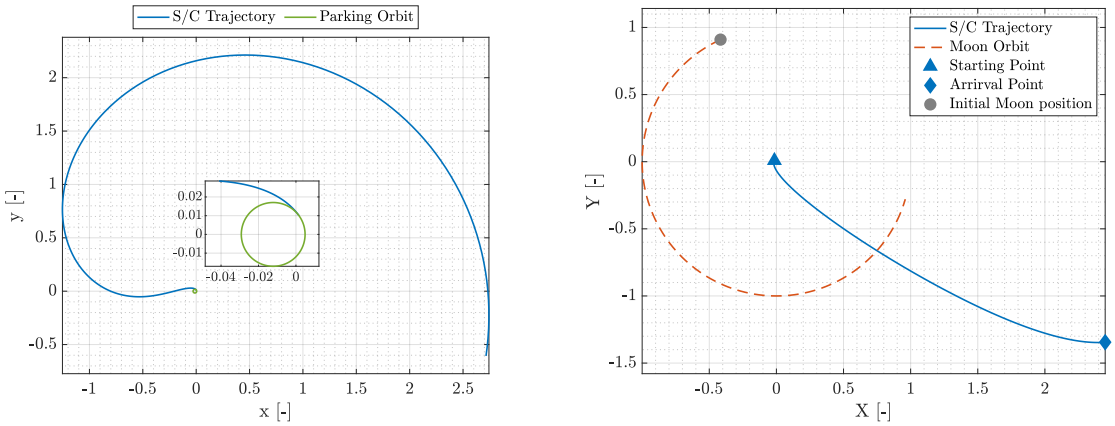
Given α, β, t_i, and δ, the vector of the initial guess can be computed using the procedure described in the reference article [1]. The radii and gravitational parameters of both the Earth and the Moon were computed using the `cspice_bodvr` MATLAB tool.
The adimensional mass parameter governing the Planar Bicircular Restricted Four-Body Problem (PBRFBP) was then calculated as $\mu = m_M/(m_E + m_M)$. The adimensional radius of the parking orbit and the initial velocity are given by $r_0 = r_i = (h_i + R_E)/DU$ and $v_0 = \beta\sqrt{(1-\mu)/r_0}$, respectively. The initial state was then defined as $\mathbf{x}_0(\alpha, \beta) = (x_0, y_0, \dot{x}_0, \dot{y}_0)$, where:

$$x_0 = r_0 \cos\alpha - \mu, \quad y_0 = r_0 \sin\alpha, \quad \dot{x}_0 = -(v_0 - r_0)\sin\alpha, \quad \dot{y}_0 = (v_0 - r_0)\cos\alpha.$$

The resulting initial state in the Earth-Moon adimensional rotating frame is reported in Table 6. The computed $\mathbf{x}_0$ was then propagated from t_i to $t_i + \delta$ integrating the equation ruling the PBRFBP dynamics (Appendix D) using `ode113` MATLAB integrator. The resulting orbit is exhibited in both the Earth-Moon rotating and Earth-centred inertial adimensional reference frames in Figure 4.

Table 6: Initial guess in Earth-Moon rotating frame.

$r_{x,0}$ $[DU]$	$r_{y,0}$ $[DU]$	$v_{x,0}$ $[VU]$	$v_{y,0}$ $[VU]$
0.001624	0.010008	−6.302738	8.674975



Earth-Moon rotating frame | Earth centred inertial frame

Figure 4: First guess solution trajectory

2.2 Simple Shooting

The objective of the NLP problem formulated below is to minimize the total cost $\Delta v = \Delta v_i + \Delta v_f$ required for the two-impulse transfer described in the reference article [1]. The problem was initially solved using simple shooting, with $\mathbf{y} = \{\mathbf{x}_i, t_i, t_f\}$ as variables vector. Where $\mathbf{x}_i$ represents the initial state, t_i is the initial time, and t_f is the final time. In order to fully define the optimization problem it is needed to clarify the cost function and the acting constraints. Firstly, it was imposed a linear inequality constraint to express the condition of $t_i \leq t_f$:

$$\begin{aligned} &A\mathbf{y} \leq \mathbf{b}, \\ &\text{with } A = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}, \text{ and } \mathbf{b} = 0. \end{aligned} \tag{5}$$

The initial state $\mathbf{x}_i = \{x_i, y_i, \dot{x}_i, \dot{y}_i\}$ must lie at a distance r_i from the Earth, with its velocity vector aligned to the local circular velocity. Similarly, the final state $\mathbf{x}_f = \{x_f, y_f, \dot{x}_f, \dot{y}_f\}$ must be at a distance r_f from the Moon, with its velocity vector also aligned to the local circular velocity. These conditions are verified if Equation 6 holds.

$$\begin{aligned} \boldsymbol{\psi}_i(\mathbf{x}_i) &= \begin{bmatrix} (x_i+\mu)^2 + y_i^2 - r_i^2 \\ (x_i+\mu)(\dot{x}_i - y_i) + y_i(\dot{y}_i + x_i + \mu) \end{bmatrix} = \mathbf{0}, \\ \boldsymbol{\psi}_f(\mathbf{x}_f) &= \begin{bmatrix} (x_f+\mu-1)^2 + y_f^2 - r_f^2 \\ (x_f+\mu-1)(\dot{x}_f - y_f) + y_f(\dot{y}_f + x_f + \mu - 1) \end{bmatrix} = \mathbf{0}. \end{aligned} \tag{6}$$

The final state $\mathbf{x}_f$ is determined by solving an initial value problem: $\mathbf{x}_f = \varphi(\mathbf{x}_i, t_i, t_f)$. Thus, the non-linear equality constraints are expressed by the function $\mathbf{c}$, as shown in Equation 7, while the objective function is given in Equation 8.

$$\mathbf{c}(\mathbf{y}) = \begin{bmatrix} \boldsymbol{\psi}_i(\mathbf{x}_i) \\ \boldsymbol{\psi}_f(\varphi(\mathbf{x}_i, t_i, t_f)) \end{bmatrix} \tag{7}$$

$$J(\mathbf{y}) = \Delta v(\mathbf{x}_i, t_i, t_f) = \Delta v_i(\mathbf{x}_i) + \Delta v_f(\varphi(\mathbf{x}_i, t_i, t_f)) \tag{8}$$

where

$$\begin{aligned} \Delta v_i &= \sqrt{(\dot{x}_i - y_i)^2 + (\dot{y}_i + x_i + \mu)^2} - \sqrt{\frac{1-\mu}{r_i}}, \\ \Delta v_f &= \sqrt{(\dot{x}_f - y_f)^2 + (\dot{y}_f + x_f + \mu - 1)^2} - \sqrt{\frac{\mu}{r_f}}. \end{aligned} \tag{9}$$

The bounds for the optimization variables were defined as follows:

- The initial position $\{x_i, y_i\}$ was constrained to be in a square of side $2r_i$ around the Earth, with $x_i \in [-\mu - r_i, -\mu + r_i]$ and $y_i \in [-r_i, r_i]$.
- The initial velocity components were limited to a magnitude up to the escape velocity from the Earth $v_{esc} = \sqrt{2(1-\mu)/r_i}$. So $\dot{x}_i \in [-v_{esc}, v_{esc}]$ and $\dot{y}_i \in [-v_{esc}, v_{esc}]$
- The initial time t_i was bounded to ensure all possible Earth-Moon-Sun configurations, with $t_i \in [0, -2\pi/\omega_s]$.
- The transfer duration δ was constrained to $[0, 23]$ TU ($\approx [0, 100]$ days), setting t_f between $t_{i,\min} + \delta_{\min}$ and $t_{i,\max} + \delta_{\max}$.

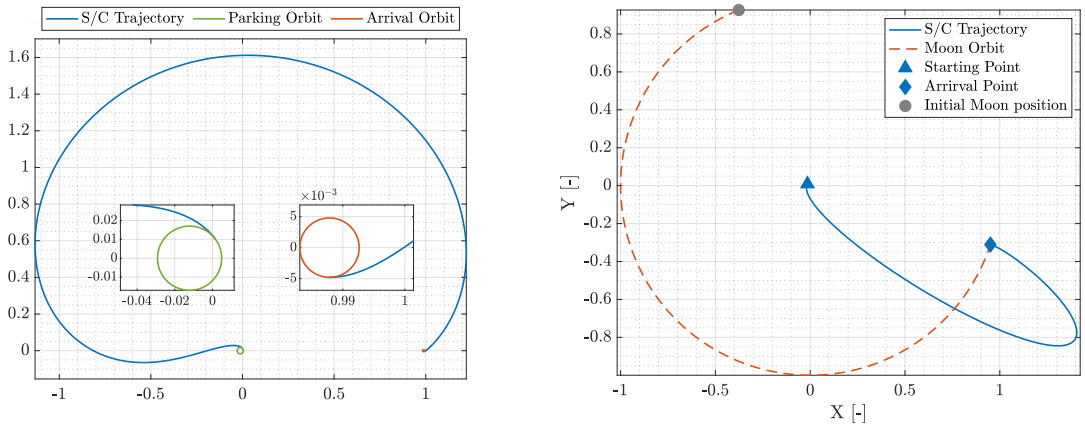
Thus, the NLP problem can be formulated as:

$$\min_{\mathbf{y}} J(\mathbf{y}) \quad \text{s.t.} \quad \begin{cases} A\mathbf{y} \leq \mathbf{b}, \\ \mathbf{c}(\mathbf{y}) = \mathbf{0}, \\ \mathbf{y}^{LB} \leq \mathbf{y} \leq \mathbf{y}^{UB}. \end{cases} \tag{10}$$

The optimization problem described above was solved using the `fmincon` MATLAB tool. Initially, the solver was used without providing any derivative. Subsequently, the analytical gradients of $J(\mathbf{y})$ and $\mathbf{c}(\mathbf{y})$ were supplied to the solver for improved performance.

2.2.1 Without providing any derivative

The NLP problem defined in Equation 10 was solved using the `active-set` algorithm. The tolerance for the equality constraint was set to 10^{-10}. For propagation, `ode113` was used with AbsTol and Reltol set to $2.5 \cdot 10^{-14}$. These stringent tolerances were chosen because of the high non linearity of PBRFBP. A forward finite difference scheme was adopted for gradient computation because, although the central scheme is more accurate, it requires twice as many function evaluations, making the algorithm significantly heavier. The results are reported in Table 7, and the resulting transfer arcs are shown in Figure 5.



Earth-Moon rotating frame | Earth centred inertial frame

Figure 5: Simple Shooting without specifying analytical gradients trajectory

2.2.2 Providing analytical gradients

To improve algorithm convergence, the gradients of the objective function and equality constraint were provided to `fmincon` in analytical form. The gradient of J is expressed as:

$$\nabla J = \begin{bmatrix} \frac{\partial J}{\partial \mathbf{x}_i} \\ \frac{\partial J}{\partial t_i} \\ \frac{\partial J}{\partial t_f} \end{bmatrix} = \begin{bmatrix} \frac{\partial \Delta v_i}{\partial \mathbf{x}_i} + \Phi^T(t_i, t_f) \frac{\partial \Delta v_f}{\partial \mathbf{x}_f} \\ -\left(\frac{\partial \Delta v_f}{\partial \mathbf{x}_f}\right)^T \Phi(t_i, t_f) \dot{\mathbf{x}}_i \\ \left(\frac{\partial \Delta v_f}{\partial \mathbf{x}_f}\right)^T \dot{\mathbf{x}}_f \end{bmatrix} \tag{11}$$

where

$$\begin{aligned} \frac{\partial \Delta v_i}{\partial \mathbf{x}_i} &= \frac{1}{\sqrt{(\dot{x}_i - y_i)^2 + (\dot{y}_i + x_i + \mu)^2}} \begin{bmatrix} \dot{y}_i + x_i + \mu \\ -(\dot{x}_i - y_i) \\ \dot{x}_i - y_i \\ \dot{y}_i + x_i + \mu \end{bmatrix} \\ \frac{\partial \Delta v_f}{\partial \mathbf{x}_f} &= \frac{1}{\sqrt{(\dot{x}_f - y_f)^2 + (\dot{y}_f + x_f + \mu - 1)^2}} \begin{bmatrix} \dot{y}_f + x_f + \mu - 1 \\ -(\dot{x}_f - y_f) \\ \dot{x}_f - y_f \\ \dot{y}_f + x_f + \mu - 1 \end{bmatrix} \end{aligned} \tag{12}$$

As shown in Equation 11, the computation of ∇J requires the State Transition Matrix (STM). Consequently, along with the propagation of dynamics, the initial value problem defined in

Equation 13 must also be solved. Here, $A(t)$ is the Jacobian of the dynamics, and its expression is reported in Appendix E.

$$\begin{cases} \dot{\Phi}(t_i, t) = A(t)\Phi(t_i, t), \\ \Phi(t_i, t_i) = \mathbf{I}_{4\times 4}. \end{cases} \tag{13}$$

The gradient of the equality constraints is given by:

$$\nabla \mathbf{c} = \begin{bmatrix} \nabla c_1 & \nabla c_2 & \nabla c_3 & \nabla c_4 \end{bmatrix}, \tag{14}$$

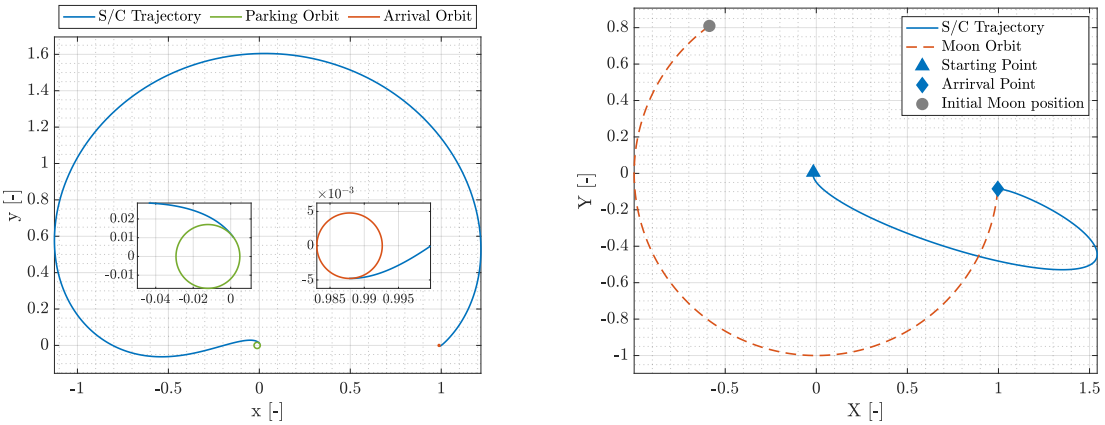
where the columns are defined as:

$$\begin{aligned} \nabla c_1 &= \begin{bmatrix} 2(x_i + \mu) & 2y_i & 0 & 0 & 0 & 0 \end{bmatrix}^T \\ \nabla c_2 &= \begin{bmatrix} \dot{x}_i & \dot{y}_i & x_i + \mu & y_i & 0 & 0 \end{bmatrix}^T \\ \nabla c_3 &= \begin{bmatrix} \Phi^T(t_i, t_f)\dfrac{\partial c_3}{\partial \mathbf{x}_f} & -\dfrac{\partial c_3}{\partial \mathbf{x}_f}\Phi^T(t_i, t_f)\dot{\mathbf{x}}_i & \left(\dfrac{\partial c_3}{\partial \mathbf{x}_f}\right)^T \dot{\mathbf{x}}_f \end{bmatrix}^T \\ \nabla c_4 &= \begin{bmatrix} \Phi^T(t_i, t_f)\dfrac{\partial c_4}{\partial \mathbf{x}_f} & -\dfrac{\partial c_4}{\partial \mathbf{x}_f}\Phi^T(t_i, t_f)\dot{\mathbf{x}}_i & \left(\dfrac{\partial c_4}{\partial \mathbf{x}_f}\right)^T \dot{\mathbf{x}}_f \end{bmatrix}^T \end{aligned}$$

with

$$\begin{aligned} \frac{\partial c_3}{\partial x_f} &= \begin{bmatrix} 2(x_f + \mu - 1) & 2y_f & 0 & 0 \end{bmatrix}^T \\ \frac{\partial c_4}{\partial x_f} &= \begin{bmatrix} \dot{x}_f & \dot{y}_f & x_f + \mu - 1 & y_f \end{bmatrix}^T \end{aligned}$$

The equality constraint tolerance and propagation tolerance were set to 10^{-10} and $2.5 \cdot 10^{-14}$, respectively, as in subsubsection 2.2.1. Propagations were performed using `ode113`, consistent with subsubsection 2.2.1. The results are presented in Table 7, and the transfer arc is shown in Figure 6.



Earth-Moon rotating frame | Earth centred inertial frame

Figure 6: Simple Shooting specifying analytical gradients trajectory

Table 7: Simple shooting solutions in the Earth-Moon rotating frame.

Gradients	$r_{x,0}$ $[DU]$	$r_{y,0}$ $[DU]$	$v_{x,0}$ $[VU]$	$v_{y,0}$ $[VU]$	t_i $[TU]$	t_f $[TU]$
False	0.001519	0.010151	-6.377743	8.588329	1.957	5.972
True	0.001620	0.010013	-6.290741	8.651415	2.198	6.203

2.3 Multiple Shooting

In the multiple shooting formulation, the solution $\mathbf{x}(t)$ of the PBRFBP equations is discretized over a uniform time grid. This results in N time nodes, equally spaced between $t_1 = t_i$ and $t_N = t_f$:

$$t_j = t_1 + \frac{j-1}{N-1}\left(t_N - t_1\right), \qquad j = 1, \ldots, N, \tag{15}$$

where the number of nodes was set to $N = 4$. Let $\mathbf{x}_j = \{x_j, y_j, \dot{x}_j, \dot{y}_j\}$ be the solution at the j-th node, i.e., $\mathbf{x}_j = \mathbf{x}(t_j)$. The 18×1 vector of variables adopted for the multiple shooting method is defined in Equation 16:

$$\mathbf{y} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 & t_1 & t_N \end{bmatrix}^T. \tag{16}$$

The objective function is then:

$$J(\mathbf{y}) = \Delta v_1 + \Delta v_2 = \Delta v_i(\mathbf{x}_1) + \Delta v_f(\mathbf{x}_N), \tag{17}$$

where Δv_i and Δv_f have forms similar to those described in Equation 9. The vector of non linear equality constraints can be written as:

$$\mathbf{c}(\mathbf{y}) = \begin{bmatrix} \boldsymbol{\zeta}_1 & \boldsymbol{\zeta}_2 & \boldsymbol{\zeta}_3 & \boldsymbol{\psi}_1 & \boldsymbol{\psi}_N \end{bmatrix}^T \tag{18}$$

where $\boldsymbol{\psi}_1$ and $\boldsymbol{\psi}_N$ are equivalent to the constraints on the initial and final condition in Simple Shooting formulation, reported in Equation 6. The defect functions, defined in Equation 19, ensure continuity between the solution at the j-th node integrated over the j-th time interval and the discretized solution at the $(j+1)$-th node.

$$\boldsymbol{\zeta}_j = \varphi(\mathbf{x}_j, t_j, t_{j+1}) - \mathbf{x}_{j+1}, \qquad j = 1, \ldots, N-1 \tag{19}$$

In the Multiple Shooting formulation, both linear and nonlinear inequality constraints are present. The linear constraint ensures that $t_1 \leq t_N$, and is expressed by Equation 20:

$$\begin{aligned} &A\mathbf{y} \leq \mathbf{b}, \text{ with} \\ &A = \begin{bmatrix} \mathbf{0}_{1\times 16} & 1 & -1 \end{bmatrix}, \text{ and } \mathbf{b} = 0. \end{aligned} \tag{20}$$

Nonlinear inequality constraints prevent the optimization process from finding solutions that result in impacts with either the Earth or the Moon:

$$\begin{aligned} \mathbf{g}(\mathbf{y}) &= \begin{bmatrix} \boldsymbol{\eta}_1 & \boldsymbol{\eta}_2 & \boldsymbol{\eta}_3 & \boldsymbol{\eta}_4 \end{bmatrix}^T < \mathbf{0}, \text{ with} \\ \boldsymbol{\eta}_j &= \begin{bmatrix} (R_E/DU)^2 - (x_j + \mu)^2 - y_j^2 \\ (R_M/DU)^2 - (x_j + \mu - 1)^2 - y_j^2 \end{bmatrix}, \quad j = 1, \ldots, N. \end{aligned} \tag{21}$$

Thus, the NLP problem in the Multiple Shooting formulation can be summarized as:

$$\min_{\mathbf{y}} J(\mathbf{y}) \text{ s.t. } \begin{cases} \mathbf{c}(\mathbf{y}) = \mathbf{0}, \\ A\mathbf{y} \leq \mathbf{b}, \\ \mathbf{g}(\mathbf{y}) < \mathbf{0}. \end{cases} \tag{22}$$

The analytical gradients of J, $\mathbf{c}$ and $\mathbf{g}$ were given to `fmincon` to improve the convergence of the algorithm. ∇J can be written as:

$$\nabla J = \begin{bmatrix} \dfrac{\partial \Delta v_1}{\partial \mathbf{x}_1}^T & \mathbf{0}_{1\times 8} & \dfrac{\partial \Delta v_2}{\partial \mathbf{x}_N}^T & 0 & 0 \end{bmatrix}^T \tag{23}$$

The expressions of $\frac{\partial \Delta v_1}{\partial \mathbf{x}_1}$ and $\frac{\partial \Delta v_2}{\partial \mathbf{x}_N}$ are equivalent to the ones present in Equation 12. The gradient of the non linear inequality constraints can be expressed as:

$$\nabla \mathbf{c} = \begin{bmatrix} \Phi(t_1,t_2) & -\mathbf{I}_{4\times4} & \mathbf{0}_{4\times4} & \mathbf{0}_{4\times4} & \frac{\partial \boldsymbol{\zeta}_1}{\partial t_1} & \frac{\partial \boldsymbol{\zeta}_1}{\partial t_N} \\ \mathbf{0}_{4\times4} & \Phi(t_2,t_3) & -\mathbf{I}_{4\times4} & \mathbf{0}_{4\times4} & \frac{\partial \boldsymbol{\zeta}_2}{\partial t_1} & \frac{\partial \boldsymbol{\zeta}_2}{\partial t_N} \\ \mathbf{0}_{4\times4} & \mathbf{0}_{4\times4} & \Phi(t_3,t_N) & -\mathbf{I}_{4\times4} & \frac{\partial \boldsymbol{\zeta}_3}{\partial t_1} & \frac{\partial \boldsymbol{\zeta}_3}{\partial t_N} \\ \frac{\partial \boldsymbol{\psi}_1}{\partial \mathbf{x}_1} & \mathbf{0}_{2\times4} & \mathbf{0}_{2\times4} & \mathbf{0}_{2\times4} & \mathbf{0}_{2\times1} & \mathbf{0}_{2\times1} \\ \mathbf{0}_{2\times4} & \mathbf{0}_{2\times4} & \mathbf{0}_{2\times4} & \frac{\partial \boldsymbol{\psi}_N}{\partial \mathbf{x}_N} & \mathbf{0}_{2\times1} & \mathbf{0}_{2\times1} \end{bmatrix} \tag{24}$$

with

$$\begin{aligned} \frac{\partial \boldsymbol{\zeta}_j}{\partial t_1} &= -\frac{N-j}{N-1}\Phi(t_j,t_{j+1})\mathbf{f}(t_j,\mathbf{x}_j) + \frac{N-j-1}{N-1}\mathbf{f}(t_j,\varphi(\mathbf{x}_j,t_j,t_{j+1})) \quad j,\dots N-1 \\ \frac{\partial \boldsymbol{\zeta}_j}{\partial t_N} &= -\frac{j-1}{N-1}\Phi(t_j,t_{j+1})\mathbf{f}(t_j,\mathbf{x}_j) + \frac{j}{N-1}\mathbf{f}(t_j,\varphi(\mathbf{x}_j,t_j,t_{j+1})) \qquad j,\dots N-1 \\ \frac{\partial \boldsymbol{\psi}_1}{\partial \mathbf{x}_1} &= \begin{bmatrix} 2(x_1+\mu) & 2y_1 & 0 & 0 \\ \dot{x}_1 & \dot{y}_1 & x_1+\mu & y_1 \end{bmatrix} \\ \frac{\partial \boldsymbol{\psi}_N}{\partial \mathbf{x}_N} &= \begin{bmatrix} 2(x_N+\mu-1) & 2y_N & 0 & 0 \\ \dot{x}_N & \dot{y}_N & x_N+\mu-1 & y_N \end{bmatrix}. \end{aligned} \tag{25}$$

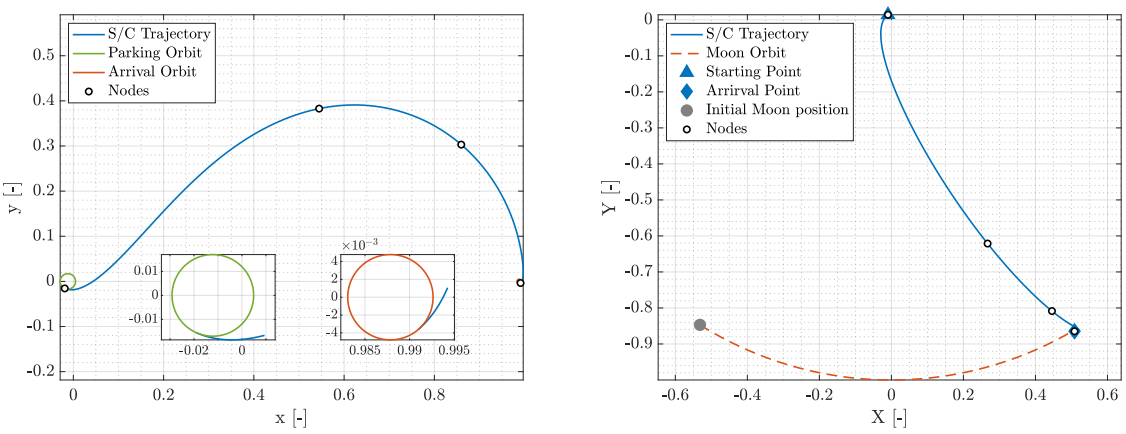
In Equation 25, $\mathbf{f}$ denotes the right-hand-side of PBRFBP equation of motion, reported in Appendix D. Finally, the gradient of the non linear inequality constraints is reported:

$$\nabla \mathbf{g} = \begin{bmatrix} \frac{\partial \boldsymbol{\eta}_1}{\partial \mathbf{x}_1} & \mathbf{0}_{2\times4} & \mathbf{0}_{2\times4} & \mathbf{0}_{2\times4} & 0 & 0 \\ \mathbf{0}_{2\times4} & \frac{\partial \boldsymbol{\eta}_2}{\partial \mathbf{x}_2} & \mathbf{0}_{2\times4} & \mathbf{0}_{2\times4} & 0 & 0 \\ \mathbf{0}_{2\times4} & \mathbf{0}_{2\times4} & \frac{\partial \boldsymbol{\eta}_3}{\partial \mathbf{x}_3} & \mathbf{0}_{2\times4} & 0 & 0 \\ \mathbf{0}_{2\times4} & \mathbf{0}_{2\times4} & \mathbf{0}_{2\times4} & \frac{\partial \boldsymbol{\eta}_4}{\partial \mathbf{x}_4} & 0 & 0 \end{bmatrix} \tag{26}$$

where:

$$\frac{\partial \boldsymbol{\eta}_1}{\partial \mathbf{x}_1} = \begin{bmatrix} -2(x_j+\mu) & -2y_j & 0 & 0 \\ -2(x_j+\mu-1) & -2y_j & 0 & 0 \end{bmatrix}. \tag{27}$$

The initial guess for the Multiple Shooting method was generated as follows. A uniform time grid, as defined in Equation 15, was created using MATLAB's `linspace` command. The initial guess from subsection 2.1 was then integrated using `ode113`, enforcing the time grid to obtain the state at $N = 4$ equally spaced time instants. The Multiple Shooting initial guess was formed as shown in Equation 16, where t_1 and t_N are the initial and final time guesses from the Simple Shooting method. The tolerances for constraints and propagations were kept consistent with those used for Simple Shooting. To ensure convergence, `MaxIterations` and `MaxFunctionEvaluations` were increased to 10000 and 40000, respectively, as `fmincon` otherwise failed to satisfy the `ConstraintTolerance` of 10^{-10} (see Appendix F). The results are presented in Table 8, and the trajectories in both the Earth-Moon rotating frame and the Earth-centered inertial frame are shown in Figure 7.



Earth-Moon rotating frame | Earth centred inertial frame

Figure 7: Multiple Shooting trajectory

Table 8: Multiple shooting solution in the Earth-Moon rotating frame.

$r_{x,0}$ $[DU]$	$r_{y,0}$ $[DU]$	$v_{x,0}$ $[VU]$	$v_{y,0}$ $[VU]$	t_i $[TU]$	t_f $[TU]$
-0.019047	-0.015567	9.752103	-4.320235	4.152	5.248

2.4 Comments on the results

In the end, it is possible to compare the results and performances of the minimization processes in Table 9. From the trajectory plots and the values of Δv and Δt, it can be observed that for the Simple Shooting method, with and without gradients, the solver converges to a very similar solution, as the optimization problem remains unchanged. However, the number of function evaluations in Table 9 highlights that providing the analytical gradient significantly improves convergence.

Table 9: Simple (SS) and Multiple Shooting (MS) with Finite Difference (FD) or Analytical Gradients (AG) performance comparison

Parameter	**SS-FD**	**SS-AG**	**MS-AG**
Δv_{tot} [km/s]	4.1069	4.1052	3.9568
Δt [days]	17.4357	17.3922	4.7584
Iter	72	15	6304
FunEval	698	31	28216
Max Constraint	9.2541e-12	5.4435e-12	1.4644e-11

As expected, the Multiple Shooting method is computationally more expensive than the Simple Shooting method. Nevertheless, the solution is notably improved in terms of both Δv and Δt. Specifically, the transfer time for the Simple Shooting solution is over three times longer than that for the Multiple Shooting solution (Table 9). A comparison between Figure 6 and Figure 7 shows that the trajectory obtained with the Multiple Shooting method is more direct, resembling the shape of a Hohmann transfer. Finally, as illustrated in Figure 8, the transfer found with the Multiple Shooting method is close to the set of transfer belonging to the Pareto front (the black lines in Figure 8), identified in the reference article [1]. This indicates that the transfer achieved with the Multiple Shooting method strikes a good balance between Δv and Δt.

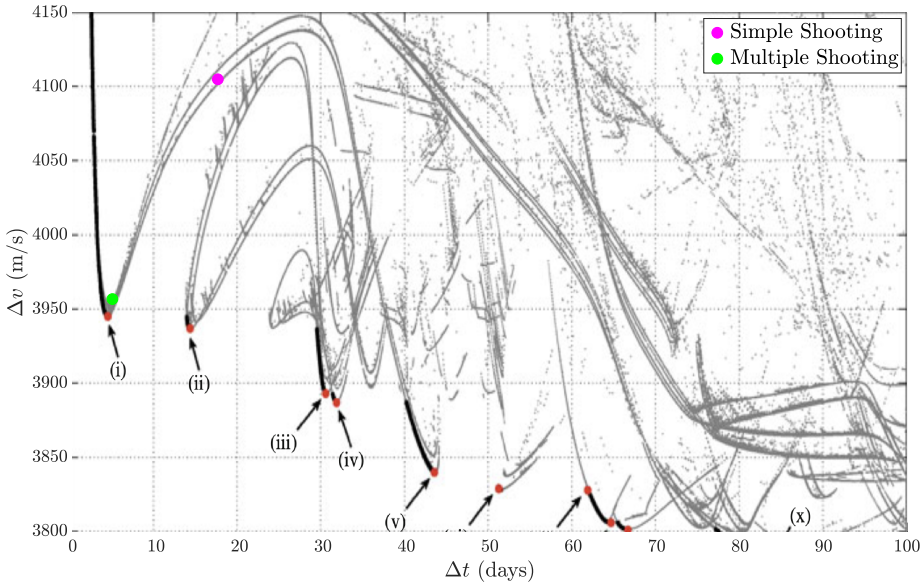


Figure 8: Optimal two-impulse Earth-Moon transfers, shown in the Δt-Δv plane. Credits: Topputo, 2013 [1]

2.5 N-body propagation

The propagation of the initial condition obtained in subsubsection 2.2.2 using the PBRFBP was compared to an n-body propagation, which accounts for a more complete gravitational model. Since the n-body propagation is not a planar problem, the initial condition was adjusted to include $z(t_i) = 0$ km and $v_z(t_i) = 0$ km/s. To perform the propagation, it was necessary to determine the initial epoch t_i, identifying when the relative positions of the Earth, Moon, and Sun matched those in subsubsection 2.2.2. This was achieved by solving a root-finding problem where the function to be zeroed is $\theta(t) - \theta_{\text{target}}$. The angle θ_{target} corresponding to the initial time t_{target} identified in subsubsection 2.2.2 can be computed as

$$\theta_{target} = \omega_s t_{target} \approx 4.2495 \text{ rad} \approx 243.4763 \text{ deg} \tag{28}$$

$\theta(t)$ is the angle between the projection of the Sun's position relative to the Earth-Moon Barycenter (EMB) onto the Earth-Moon plane and the EMB-Moon axis. $\theta(t)$ was computed following the procedure below:

- Given the ephemeris time et, use the `spk kernels` to obtain the Moon's state vector and the Sun's position vector relative to the Earth-Moon Barycenter (EMB) in the J2000 reference frame.
- Construct the rotation matrix $\mathbf{U} = \begin{bmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3\end{bmatrix}^T$, where:
  - $\mathbf{u}_1$ is the unit vector in the direction of the Moon,
  - $\mathbf{u}_3$ is the unit vector parallel to the Moon's angular momentum,
  - $\mathbf{u}_2$ is the unit vector orthogonal to both $\mathbf{u}_1$ and $\mathbf{u}_3$.
- Project the Sun's position onto the orbital plane by premultiplying its position vector by the rotation matrix $\mathbf{U}$.
- Compute $\theta(t)$ as the angle between the projected Sun vector and the Earth-Moon axis.

The initial epoch was searched between 2024 Sep 28 and 2024 Oct 25. This time span was chosen because the Moon's orbital period around the Earth is approximately 27 days, allowing the range $[0, 2\pi]$ of all possible θ to be covered. The described zero-finding problem was solved using MATLAB's `fsolve`, with an `OptimalityTolerance` of 10^{-12}. The initial guess was selected based on the graph shown in Figure 9. The final epoch was obtained by adding the transfer time calculated in subsubsection 2.2.2 to the initial epoch.

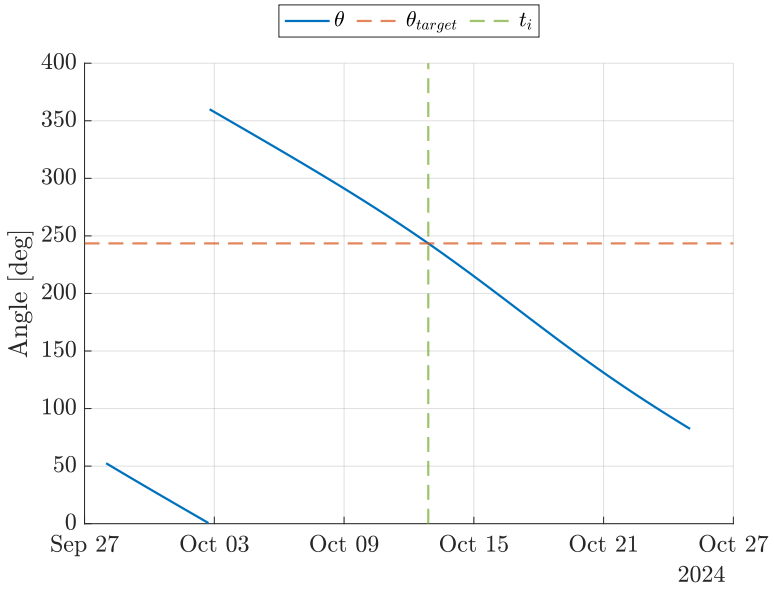


Figure 9: Time evolution of θ

The results, including t_i, t_f, and $\mathbf{x}_0$, are reported in Table 10. The trajectories, in the Earth-centered ECLIPJ2000 reference frame, obtained with the PBRFBP and the n-body propagation are shown in Figure 10. It can be observed that, initially, the two trajectories almost overlap. However, over time, the n-body trajectory begins to deviate from the x-y plane due to the influence of forces not accounted for in the PBRFBP model.

Table 10: Initial epoch, final epoch, and initial state in Earth-centered ECLIPJ2000 frame.

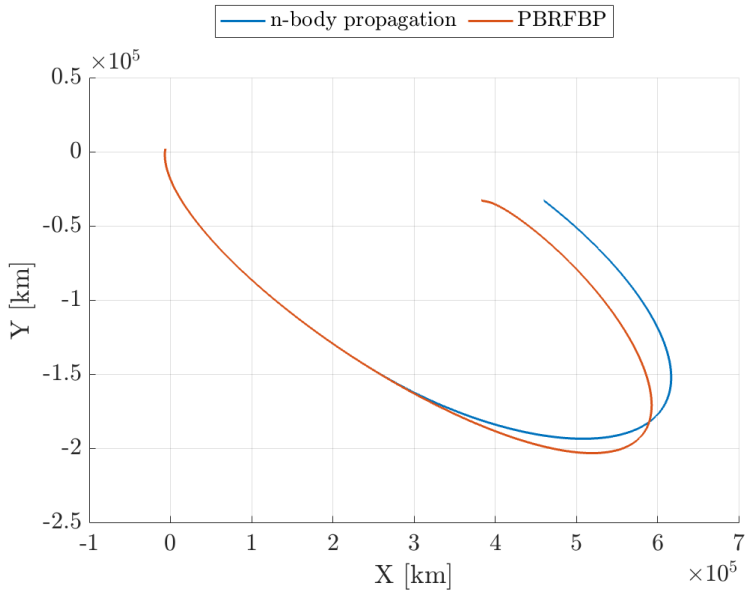
Symbol	**Calendar epoch** (UTC)
t_i	2024-10-12T21:25:33.835
t_f	2024-10-30T06:50:19.993

Times

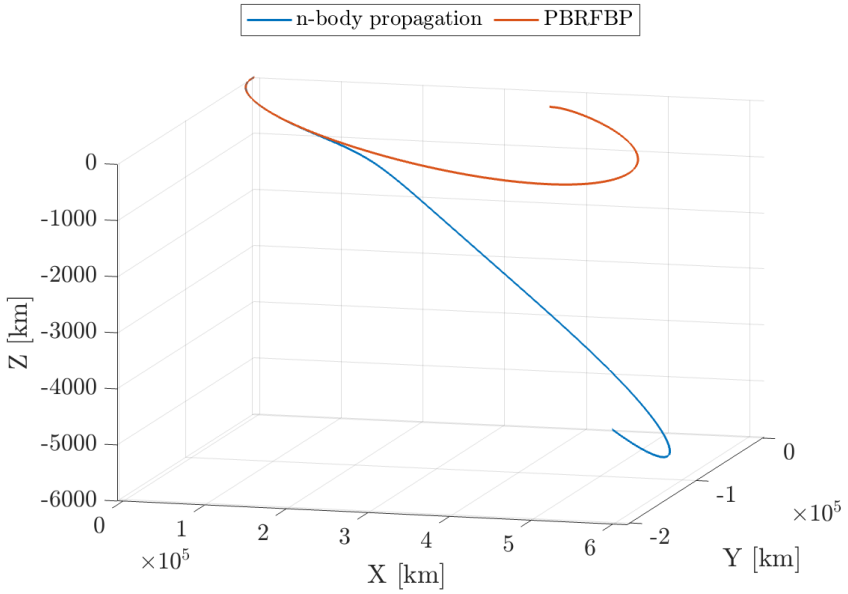
$r_{x,0}$ $[km]$	$r_{y,0}$ $[km]$	$r_{z,0}$ $[km]$
-6223.62888303	2026.14329181	0.0

$v_{x,0}$ $[km/s]$	$v_{y,0}$ $[km/s]$	$v_{z,0}$ $[km/s]$
-3.39799637	-10.43749888	0.0

Initial State



top view of PBRFBP and n-body trajectories in Earth-centered ECLIPJ2000 reference frame



3D view of PBRFBP and n-body trajectories in Earth-centered ECLIPJ2000 reference frame

Figure 10: It is important to note that, in the 3D view, the scale of the Z axis is two orders of magnitude smaller than that of the X and Y axes. This choice of using different scale for z-axis was made to ensure that the non-planarity of the n-body propagation could be appreciated, as it would otherwise be difficult to discern.

3 Continuous guidance

Exercise 3

A low-thrust option is being considered to perform an orbit raising maneuver using a low-thrust propulsion system in Earth orbit. The spacecraft is released on a circular orbit on the equatorial plane at an altitude of 800 km and has to reach an orbit inclined by 0.75 deg on the equatorial plane at 1000 km. This orbital regime is characterized by a large population of resident space objects and debris, whose spatial density q can be expressed as:

$$q(\rho) = \frac{k_1}{k_2 + \left(\dfrac{\rho - \rho_0}{DU}\right)^2}$$

where ρ is the distance from the Earth center. The objective is to design an optimal orbit raising that minimizes the risk of impact, that is to minimize the following objective function

$$F(t) = \int_{t_0}^{t_f} q(\rho(t))\,\mathrm{d}t.$$

The parameters and reference Distance Unit to be considered are provided in Table 11.

Symbol	Value	Units	Meaning
h_i	800	km	Altitude of departure orbit
h_f	1000	km	Altitude of arrival orbit
Δi	0.75	deg	Inclination change
R_e	6378.1366	km	Earth radius
μ	398600.435	$\mathrm{km}^3/\mathrm{s}^2$	Earth gravitational parameter
ρ_0	$750 + R_e$	km	Reference radius for debris flux
k_1	1×10^{-5}	DU^{-1}	Debris spatial density constant 1
k_2	1×10^{-4}	DU^2	Debris spatial density constant 2
m_0	1000	kg	Initial mass
$T_{\max}$	3.000	N	Maximum thrust
I_{sp}	3120	s	Specific impulse
DU	7178.1366	km	Distance Unit
MU	m_0	kg	Mass Unit

Table 11: Problem parameters and constants. The units of time TU and velocity VU can be computed imposing that the scaled gravitational parameter $\overline{\mu} = 1$.

1) Plot the debris spatial density $q(\rho) \in [h_i - 100;\ h_f + 100]$ km and compute the initial state and target orbital state, knowing that: i) the initial and final state are located on the x-axis of the equatorial J2000 reference frame; ii) the rotation of the angle Δi is performed around the x-axis of the equatorial J2000 reference frame (RAAN = 0).

2) Adimensionalize the problem using as reference length $DU = \rho_i = h_i + R_e$ and reference mass $MU = m_0$, imposing that $\mu = 1$. Report all the adimensionalized parameters.

3) Using the PMP, write down the spacecraft equations of motion, the costate dynamics, and the zero-finding problem for the unknowns $\{\boldsymbol{\lambda}_0, t_f\}$ with the appropriate transversality condition. **Hint**: the spacecraft has to reach the target state computed in point 1).

4) Solve the problem considering the data provided in Table 11. To obtain an initial guess for the costate, generate random numbers such that $\lambda_{0,i} \in [-250; +250]$ while $t_f \approx 20\pi$. Report the obtained solution in terms of $\{\boldsymbol{\lambda}_0, t_f\}$ and the error with respect to the target.

Assess your results exploiting the properties of the Hamiltonian in problems that are not time-dependent and time-optimal solution. Plot the evolution of the components of the primer vector $\boldsymbol{\alpha}$ in a NTW reference frame[†].

5) Solve the problem for a lower thrust level $T_{\max} = 2.860$ N. Compare the new solution with the one obtained in the previous point. **Hint**: exploit numerical continuation. (11 points)

[†]The T-axis is aligned with the velocity, the N-axis is parallel to the angular momentum, while the W-axis is pointing inwards, i.e., towards the Earth.

3.1 Debris density and boundary states

The density function $q(\rho)$ described in the exercise text is shown in the interval $h \in [700, 1100]$ km (Figure 11).

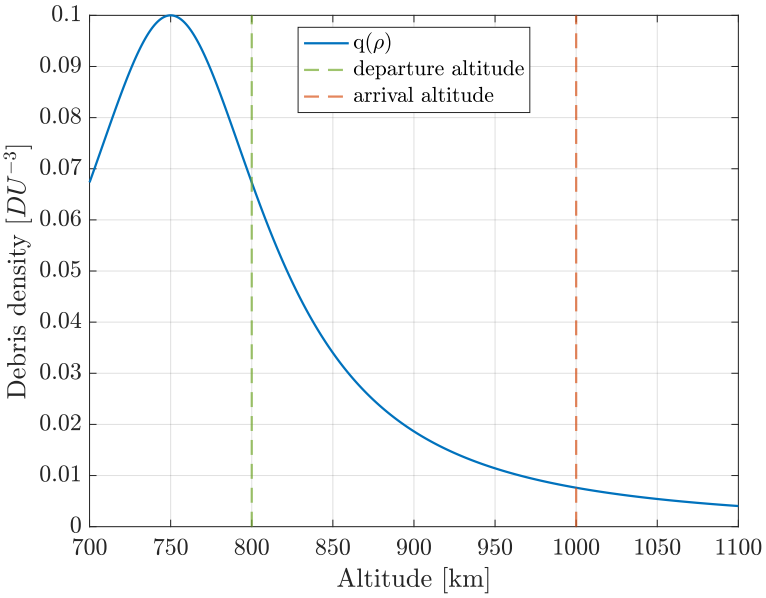


Figure 11: Debris density.

Given the altitude at release h_i and the altitude of the target orbit h_f, and knowing that the two orbits are circular, the magnitude of the position and velocity can be computed as:

$$r_i = R_E + h_i, \quad r_f = R_E + h_f, \tag{29}$$

$$v_i = \sqrt{\frac{\mu}{r_i}}, \quad v_f = \sqrt{\frac{\mu}{r_f}}. \tag{30}$$

The release orbit is equatorial, thus the position is aligned with the x-axis and the velocity with the y-axis. Therefore, the initial state can be computed as shown in Equation 31:

$$\mathbf{x}_i = \begin{bmatrix} r_i & 0 & 0 & v_i & 0 & 0 & m_0 \end{bmatrix}^T, \tag{31}$$

where m_0 is the initial mass of the spacecraft (S/C). Knowing that the second orbit is inclined by $\Delta i = 0.75°$ with respect to the equator, the target Cartesian state is:

$$\mathbf{rv}_f = \begin{bmatrix} r_f & 0 & 0 & 0 & v_f \cos \Delta i & v_f \sin \Delta i \end{bmatrix}^T. \tag{32}$$

The initial and target Cartesian states are reported in Table 12.

Table 12: Initial and target state in Earth-centered equatorial J2000 inertial frame.

$r_{x,i}$ $[km]$	$r_{y,i}$ $[km]$	$r_{z,i}$ $[km]$	$v_{x,i}$ $[km/s]$	$v_{y,i}$ $[km/s]$	$v_{z,i}$ $[km/s]$
7178.136600	0000.000000	0000.000000	0.00000000	7.45183148	0.00000000
$r_{x,f}$ $[km]$	$r_{y,f}$ $[km]$	$r_{z,f}$ $[km]$	$v_{x,f}$ $[km/s]$	$v_{y,f}$ $[km/s]$	$v_{z,f}$ $[km/s]$
7378.136600	0000.000000	0000.000000	0.00000000	7.34950906	0.0962103438

3.2 Adimensionalization of variables

Although the reference length and the reference mass were provided by the text, the reference time (TU) needed for adimensionalizing the problem must be computed. Observing the units of μ, $[L]^3\,[T]^{-2}$, the reference time can be calculated by imposing $\overline{\mu} = 1$ as:

$$TU = \sqrt{\frac{DU^3}{\mu}},$$

where DU is expressed in km and μ in $\mathrm{km}^3/\mathrm{s}^2$. From TU, the reference velocity (VU) and the reference force (FU) can also be derived as follows:

$$VU = \frac{DU}{TU}, \quad FU = \frac{MU \cdot VU}{TU} \cdot 10^3.$$

The factor 10^3 is necessary to ensure that the resulting unit of the reference force is in N rather than kN. If omitted, the incorrect scaling would lead to a wrong adimensionalization of the thrust, as it is given in N. Similarly, a 10^{-3} factor is included in the adimensionalization of g_0, because the reference distance is expressed in km, while the unit of g_0 is in $\mathrm{m/s}^2$. Table 13 reports the adimensionalized parameters, while Table 14 lists the reference quantities used for the adimensionalization. In Table 13, the $\overline{(\cdot)}$ notation denotes the adimensionalized form of a corresponding dimensional quantity.

Table 13: Adimensional parameters

$\overline{\mathbf{r}}_i$	1	0	0
$\overline{\mathbf{v}}_i$	0	1	0
$\overline{m}_i$	1		
$\overline{\mathbf{r}}_f$	1.0279	0	0
$\overline{\mathbf{v}}_f$	0	0.9862	0.0129
$\overline{\mu}$	1		
$\overline{I}_{sp}$	3.2390		
$\overline{T}_{max}$	3.8779e-4		
$\overline{g}_0$	1.2681		
$\overline{\rho}_0$	0.9930		

Table 14: Reference quantities

DU	7178.1366 km
MU	1000 kg
TU	963.2714 s
VU	7.4518 km/s
FU	7735.9620 N

3.3 Statement of the problem

The optimal orbit raising described in the exercise can be formalized as an optimal control problem (OCP), as shown in Equation 33:

$$\min_{(\hat{\boldsymbol{\alpha}},u)\in\Omega} J(\mathbf{x},\mathbf{u},t) = \int_{t_0}^{t_f} q(\rho(t))\,\mathrm{d}t \quad \text{s.t.} \quad \begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{u},t), \\ \mathbf{x}(t_0) = \mathbf{x}_0, \\ \mathbf{r}(t_f) = \mathbf{r}_f, \\ \mathbf{v}(t_f) = \mathbf{v}_f, \\ \lambda_m(t_f) = 0. \end{cases} \tag{33}$$

Here, $q(\rho(t))$ is the debris density function, $\mathbf{f}$ is the right-hand side of Equation 34, and Ω is the set of admissible control actions: $\Omega = \{(\hat{\boldsymbol{\alpha}}, u) : \hat{\alpha} = 1, u \in [0,1]\}$. The condition $\lambda_m(t_f) = 0$ is present since $m(t_f)$ cannot be determined a-priori.

The dynamics of a spacecraft moving in the two-body problem while generating thrust with its engines is described by Equation 34:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\overline{\mathbf{r}}} \\ \dot{\overline{\mathbf{v}}} \\ \dot{\overline{m}} \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{v}} \\ -\dfrac{\overline{\mu}}{\overline{r}^3}\overline{\mathbf{r}} + u\dfrac{\overline{T}_{\max}}{\overline{m}}\hat{\boldsymbol{\alpha}} \\ -u\dfrac{\overline{T}_{\max}}{\overline{I}_{\mathrm{sp}}\overline{g}_0} \end{bmatrix}. \tag{34}$$

The Hamiltonian is defined as $H = q(\rho) + \boldsymbol{\lambda}\cdot\mathbf{f}$, with the explicit expression given by:

$$H = q(\rho) + \boldsymbol{\lambda}_r\cdot\overline{\mathbf{v}} - \frac{\overline{\mu}}{\overline{r}^3}\overline{\mathbf{r}}\cdot\boldsymbol{\lambda}_v + u\frac{\overline{T}_{\max}}{\overline{m}}\boldsymbol{\lambda}_v\cdot\hat{\boldsymbol{\alpha}} - \lambda_m u\frac{\overline{T}_{\max}}{\overline{I}_{\mathrm{sp}}\overline{g}_0} \tag{35}$$

The Pontryagin Maximum Principle (PMP) states that the admissible control action $\mathbf{u}\in\Omega$ that minimizes the functional J described in Equation 33 is the one that minimizes the Hamiltonian:

$$\mathbf{u}^* = (\hat{\boldsymbol{\alpha}}^*, u^*) \coloneqq \underset{(\hat{\boldsymbol{\alpha}}^*,u^*)\in\Omega}{\arg\min}\ H(\mathbf{x},\boldsymbol{\lambda},\mathbf{u}) \tag{36}$$

Looking at the Hamiltonian expression in Equation 35, it is possible to note that the thrust direction that minimizes the Hamiltonian is the Primer Vector, which is defined as:

$$\hat{\boldsymbol{\alpha}}^* = -\frac{\boldsymbol{\lambda}_v}{\lambda_v} \tag{37}$$

Substituting $\hat{\boldsymbol{\alpha}}^*$ into the Hamiltonian, we get:

$$H = q(\rho) + \boldsymbol{\lambda}_r\cdot\overline{\mathbf{v}} - \frac{\overline{\mu}}{\overline{r}^3}\overline{\mathbf{r}}\cdot\boldsymbol{\lambda}_v + \frac{\overline{T}_{\max}}{\overline{I}_{\mathrm{sp}}\overline{g}_0}uS_t, \tag{38}$$

where $S_t = -\dfrac{\lambda_v}{\overline{m}}\overline{I}_{\mathrm{sp}}\overline{g}_0 - \lambda_m$ is the switching function. The sign of S_t determines the control policy and, consequently, the time history of the throttling factor u. The factor $\dfrac{\lambda_v}{\overline{m}}\overline{I}_{\mathrm{sp}}\overline{g}_0$ is positive, as it consists of positive quantities.

$$\dot{\lambda}_m = -\frac{\partial H}{\partial \overline{m}} = -u\frac{\overline{T}_{\max}\lambda_v}{\overline{m}^2} \tag{39}$$

λ_m is always positive, since it must be zero at t_f and, since $u \in [0,\ 1]$, its time derivative, shown in Equation 39, is always ≤ 0. This implies that S_t is negative, and the throttling factor that minimizes the Hamiltonian H is $u^* = 1$.

Finally, the Euler-Lagrange equations, which govern the state and costate dynamics, can be written, embedding $\hat{\boldsymbol{\alpha}}^*$ and u^*:

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\boldsymbol{\lambda}} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial H}{\partial \boldsymbol{\lambda}} \\ -\dfrac{\partial H}{\partial \mathbf{x}} \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{v}} \\ -\dfrac{\overline{\mu}}{\overline{r}^3}\overline{\mathbf{r}} - \dfrac{\overline{T}_{\max}}{\overline{m}\lambda_v}\boldsymbol{\lambda}_v \\ -\dfrac{\overline{T}_{\max}}{\overline{I}_{\mathrm{sp}}\overline{g}_0} \\ \left(\dfrac{2k_1\left(\overline{\rho}-\overline{\rho}_0\right)}{\left[k_2+\left(\overline{\rho}-\overline{\rho}_0\right)^2\right]^2}\right)\dfrac{\overline{\mathbf{r}}}{\overline{r}} - \dfrac{3\overline{\mu}}{\overline{r}^5}\left(\overline{\mathbf{r}}\cdot\boldsymbol{\lambda}_v\right)\overline{\mathbf{r}} + \dfrac{\overline{\mu}}{\overline{r}^3}\boldsymbol{\lambda}_v \\ -\boldsymbol{\lambda}_r \\ -\dfrac{\overline{T}_{\max}\lambda_v}{\overline{m}^2} \end{bmatrix}. \tag{40}$$

The boundary conditions that have to be satisfied are reported in Equation 41. Since the final time is not fixed, to fully define the Two Point Boundary Value Problem (TPBVP), it is necessary to impose the transversality condition. In this case, it assumes a particularly simple form since the target state is fixed (Equation 42).

$$\begin{cases} \mathbf{x}(t_0) = \mathbf{x}_0 \\ \overline{\mathbf{r}}(t_f) = \overline{\mathbf{r}}_f \\ \overline{\mathbf{v}}(t_f) = \overline{\mathbf{v}}_f \\ \lambda_m(t_f) = 0 \end{cases} \tag{41}$$

$$H(t_f) = 0 \tag{42}$$

The so defined TPBVP can be reformulated as a zero-finding problem: find $\mathbf{y} = [\boldsymbol{\lambda}_0, t_f]^T$ such that $\varphi\left(\begin{bmatrix}\mathbf{x}_0 \\ \boldsymbol{\lambda}_0\end{bmatrix}, t_0, t_f\right)$ produces $\mathbf{x}(t_f)$ and $\boldsymbol{\lambda}(t_f)$ that satisfy Equation 43:

$$\mathbf{F}\left(\mathbf{y}\right) = \begin{bmatrix} \overline{\mathbf{r}}\left(t_f\right) - \overline{\mathbf{r}}_f \\ \overline{\mathbf{v}}\left(t_f\right) - \overline{\mathbf{v}}_f \\ \lambda_m\left(t_f\right) \\ H\left(t_f\right) \end{bmatrix} = \mathbf{0}_{4\times 1}. \tag{43}$$

3.4 Solution for T=3 N

Algorithm 2 was implemented to solve the zero-finding problem described in Equation 43. Since the costate components lack physical meaning, it was necessary to generate $\boldsymbol{\lambda}_0^{(0)}$ as a set of random numbers at each step of Algorithm 2. The first six components were chosen within the range ± 250. Given that λ_m is a non-negative function, as discussed in subsection 3.3, $\lambda_{0m}^{(0)}$ was selected from the interval $[0, 250]$. The initial guess for $t_f^{(0)}$ was obtained by perturbing the nominal value of 20π with a random value in the interval $\pm\pi$.

After generating the initial guess, Equation 43 was solved using MATLAB's `fsolve` function with the *Levenberg-Marquardt* algorithm. A `FunctionTolerance` of 10^{-7} was specified, corresponding to approximately 0.72 m, which aligns with the typical dimensions of spacecraft. Additionally, the `MaxFunctionEvaluations` and `MaxIterations` parameters were increased to 20000 and 1000, respectively, while the `StepTolerance` was tightened to 10^{-10}.

Algorithm 2 λ_0

Ensure: Adjusted $\boldsymbol{\lambda}_0$, t_f $\boldsymbol{\lambda}_0^{(0)}$, $t_f^{(0)}$
1: Initialize $F_{opt} = 1$ and iter $= 0$
2: **while** exitflag < 0 **and** iter $< N_{max}$ **do**
3: Guess $\boldsymbol{\lambda}_0^{(0)}$ and $t_f^{(0)}$
4: Solve $\mathbf{F}(\mathbf{y}) = \mathbf{0}$
5: **if** $F < F_{opt}$ **then**
6: $F_{opt} = F \quad \mathbf{y}_{opt} = \mathbf{y}$
7: **end if**
8: $iter = iter + 1$
9: **end while**

Derivatives were computed using a central finite difference scheme, which offers higher accuracy compared to the forward difference approach. The propagation of Equation 40, required to compute errors in Equation 43, was performed using the `ode113` solver with both `RelTol` and `AbsTol` set to $2.5 \cdot 10^{-14}$. The `while` loop in Algorithm 2 is necessary because the convergence is strongly influenced by the choice of the initial guess, which, as explained earlier, is generated randomly. Despite the stringent settings, the solver stalled when converging. Specifically, the algorithm terminated due to the magnitude of the difference between two successive steps falling below the specified threshold, without achieving the desired `FunctionTolerance`. When this occurred, Algorithm 2 was re-executed, generating new initial guesses by perturbing the outputs of the previous run by a certain percentage, initially set to 10%. This procedure was repeated multiple times, continuously reducing the percentage of perturbation, until reaching a solution with an acceptable residual. This procedure enabled achieving errors much smaller than the specified tolerance, as illustrated in Table 15.

Table 15: Final state error with respect to target position and velocity ($T_{\max} = 3.000$ N).

Error	Value	Units
$\|\|\mathbf{r}(t_f) - \mathbf{r}_f\|\|$	1.5644e-8	km
$\|\|\mathbf{v}(t_f) - \mathbf{v}_f\|\|$	1.5158e-8	m/s

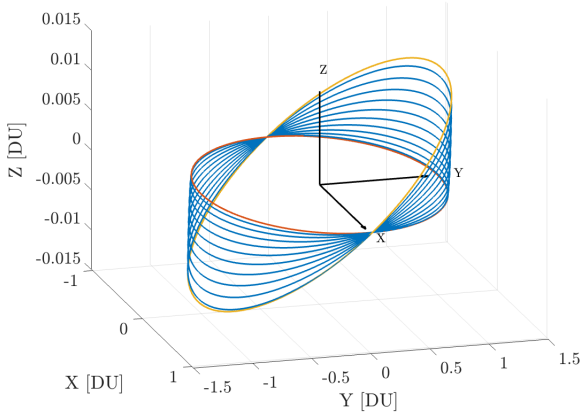


Figure 12: Orbit raising with $T = 3$ N in the Earth-centered equatorial J2000 reference frame

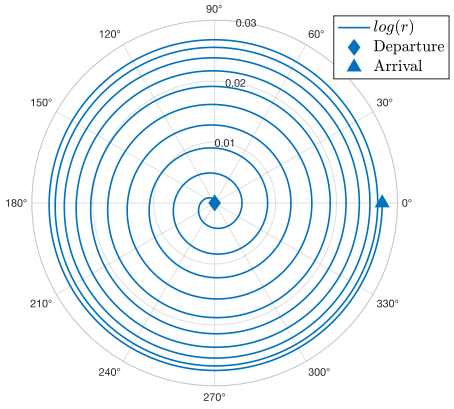


Figure 13: Polar plot of the raising

The obtained results for $\boldsymbol{\lambda}_0$, m_f, and t_f are presented in Table 16. The orbit-raising trajectory is depicted in Figure 12. The polar plot in Figure 13 illustrates the logarithm of the distance from Earth in polar coordinates. The resulting spiral exhibits more widely spaced spires near the

origin, indicating that most of the orbit-raising occurs during the initial phase of the transfer. This result is consistent with the shape of the debris density distribution shown in Figure 11.

Table 16: Optimal orbit raising transfer solution ($T_{\max} = 3.000$ N).

t_f [mins]	m_f $[kg]$
1035.1974	993.9120

λ_{0,r_x}	λ_{0,r_y}	λ_{0,r_z}	λ_{0,v_x}	λ_{0,v_y}	λ_{0,v_z}	$\lambda_{0,m}$
-214.9812	-10.3659	0.8856	-10.3929	-214.6105	-112.9454	2.5964

Indeed, between r_i and r_f, the debris density is a strictly decreasing function and, to minimize the probability of collisions, it is logical that the optimal trajectory ensures the spacecraft spends as little time as possible in the regions where the debris density is high.

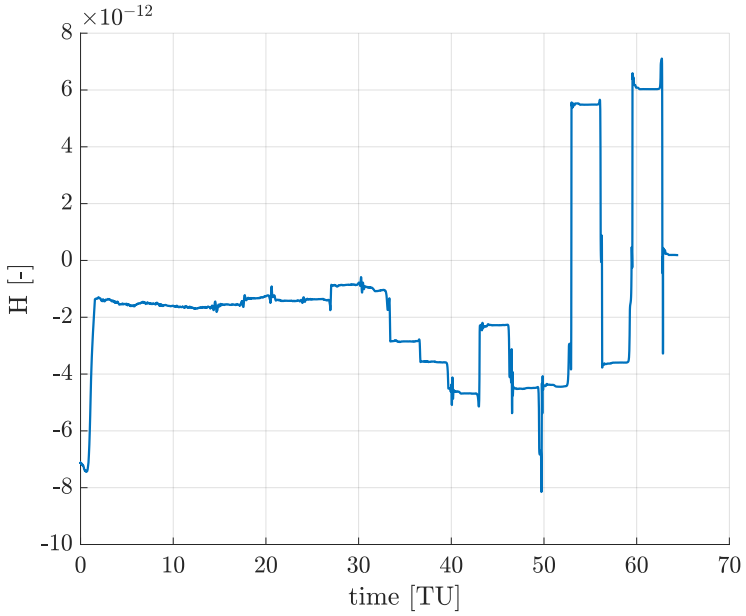


Figure 14: Time history of the hamiltonian

Figure 14 shows that, apart from some oscillations due to numerical errors, the Hamiltonian remains equal to zero throughout the entire transfer. This behavior was expected since H is forced to be equal to zero at the final time due to the transversality condition. Furthermore, for time-independent problems, the Hamiltonian is constant over time, which implies it must be zero for the entire transfer.

Figure 15 illustrates the time history of the $\hat{\boldsymbol{\alpha}}^*$ components in the NTW reference frame, where N is aligned with the angular momentum, T is parallel to the velocity, and W points inward, i.e., towards the Earth. The vertical bars indicate the time instants when the spacecraft (S/C) crosses the equatorial plane, corresponding to $z = 0$. From the graph of $\hat{\alpha}_N^*$ in Figure 15, it can be observed that the maxima and minima coincide with the nodal line crossings. This result is logical because the departure and arrival orbits share the same right ascension of the ascending node ($RAAN$), and the only way to change the orbital plane without altering the $RAAN$ is to perform the plane change maneuver at the nodes.

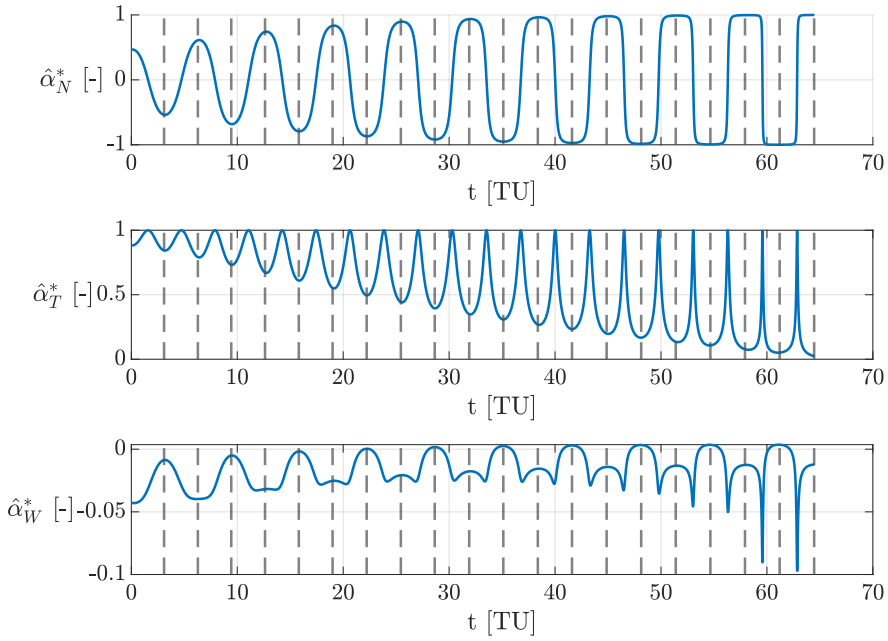


Figure 15: Time history of Primer Vector

It can also be noted that $\hat{\alpha}_N^*$ oscillates with increasing amplitude. This behavior occurs because, as previously explained, the orbit raising is primarily completed during the initial phase of the transfer. Since $\hat{\alpha}^* = 1$ at all times, a high magnitude of the in-plane components of $\hat{\boldsymbol{\alpha}}^*$ results in a smaller out-of-plane component $\hat{\alpha}_N^*$. Furthermore, $\hat{\alpha}_T^*$ and $\hat{\alpha}_W^*$ are consistently positive and negative, respectively. This indicates that the in-plane components of $\hat{\boldsymbol{\alpha}}^*$ are always directed towards the velocity and the positive radial direction, which is consistent with an orbit-raising maneuver.

3.5 Numerical continuation

To solve Equation 43 for the thrust level of $T_{max} = 2.860$ N while ensuring fast convergence of the algorithm, numerical continuation was employed. The results for $\boldsymbol{\lambda}_0$ and t_f obtained for $T_{max} = 3$ N, as reported in Table 16, were used as the initial guess to solve Equation 43 for $T_{max} = 2.965$ N. This process was repeated, gradually decreasing the thrust level, until the solution for $T_{max} = 2.860$ N was found. At each step, the solution obtained in the previous step served as the initial guess for the next step. The `OptimalityTolerance` and `FunctionTolerance` were set to 10^{-10} and 10^{-8}, respectively. These stringent values were chosen to limit the propagation of numerical errors during the continuation process. As shown in Table 17, the errors are of the same order of magnitude as those obtained for $T_{max} = 3$ N, reported in Table 15.

Table 17: Final state error with respect to target position and velocity ($T_{\max} = 2.860$ N).

Error	Value	Units
$\|\|\mathbf{r}(t_f) - \mathbf{r}_f\|\|$	3.7569e-8	km
$\|\|\mathbf{v}(t_f) - \mathbf{v}_f\|\|$	3.7226e-8	m/s

The results obtained for $T_{max} = 2.860$ N are reported in Table 18. It can be noted that, counterintuitively, the time of flight for $T_{max} = 2.860$ N is approximately 5 min shorter than that obtained for $T_{max} = 3$ N (see Table 16). Consequently, the consumed mass is also smaller.

Table 18: Optimal orbit raising transfer solution ($T_{\max} = 2.860$ N).

t_f [mins]	m_f [kg]
1030.9760	994.2198

λ_{0,r_x}	λ_{0,r_y}	λ_{0,r_z}	λ_{0,v_x}	λ_{0,v_y}	λ_{0,v_z}	$\lambda_{0,m}$
-593.1529	-11.5904	2.2972	-11.9293	-592.7890	-920.3698	17.3814

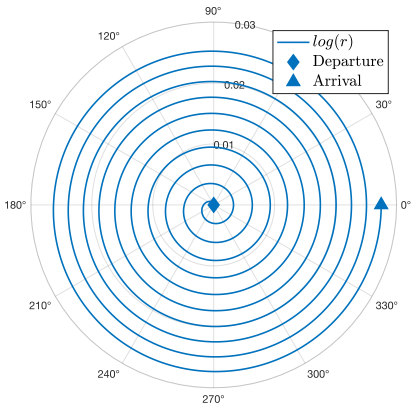


Figure 16: Polar plot for $T_{max} = 2.860$ N

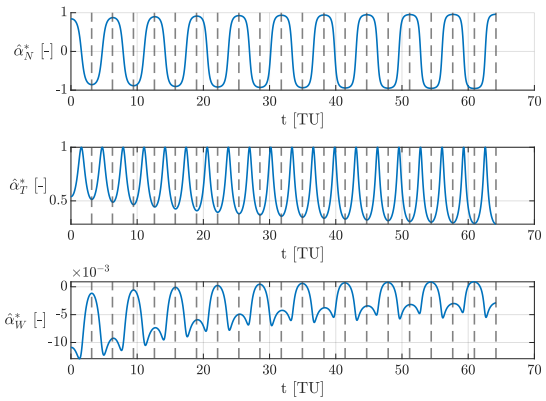


Figure 17: Time history of $\hat{\boldsymbol{\alpha}}^*$ for $T_{max} = 2.860$ N

As shown in Figure 16, the transfer is more regular than in the $T_{max} = 3$ N case, and the behavior of $r(t)$, as depicted in Figure 13, is less pronounced. However, from Figure 17, it can still be observed that α_T^* and α_W^* have slightly greater magnitudes during the initial phase of the transfer. This behavior is consistent with the shape of the debris density distribution shown in Figure 11. Additionally, Figure 17 indicates that the plane change is primarily performed around the nodes, and the signs of the in-plane components of $\hat{\boldsymbol{\alpha}}^*$ are consistent with an orbit-raising maneuver.

A CRTBP potential function

$$\Omega(x, y, z) = \frac{1}{2}(x^2 + y^2) + \frac{1-\mu}{r_1} + \frac{\mu}{r_2} + \frac{1}{2}\mu(1-\mu)$$

The function $\Omega(x, y, z)$ represents the effective potential in the Circular Restricted Three-Body Problem (CRTBP), where the first term accounts for the centrifugal potential, and the other terms describe the gravitational potentials from the two primary bodies and a constant term related to the system's mass ratio.

where $r_1 = \sqrt{(x+\mu)^2 + y^2 + z^2}$ and $r_2 = \sqrt{(x+\mu-1)^2 + y^2 + z^2}$ The distances r_1 and r_2 are the Euclidean distances from the spacecraft to the two primary bodies.

B CRTBP dynamics

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}) = \begin{bmatrix} v_x \\ v_y \\ v_z \\ 2v_y + x - \frac{(1-\mu)(x+\mu)}{r_1^3} - \frac{\mu(x+\mu-1)}{r_2^3} \\ -2v_x - \frac{y(1-\mu)}{r_1^3} - \frac{\mu y}{r_2^3} \\ -\frac{z(1-\mu)}{r_1^3} - \frac{\mu z}{r_2^3} \end{bmatrix}$$

The equations describe the motion of a spacecraft in the Circular Restricted Three-Body Problem (CRTBP).

where $r_1 = \sqrt{(x+\mu)^2 + y^2 + z^2}$ and $r_2 = \sqrt{(x+\mu-1)^2 + y^2 + z^2}$.

C Jacobi Constant

$$J(x, y, z, v_x, x_y, v_z) \coloneqq 2\Omega(x, y, z) - v^2 = C$$

where $r_1 = \sqrt{(x+\mu)^2 + y^2 + z^2}$ and $r_2 = \sqrt{(x+\mu-1)^2 + y^2 + z^2}$ and

$\Omega(x, y, z) = \frac{1}{2}(x^2 + y^2) + \frac{1-\mu}{r_1} + \frac{\mu}{r_2} + \frac{1}{2}\mu(1-\mu)$ with $v^2 = v_x^2 + v_y^2 + v_z^2$.

D PBRFBP dynamics

The equations describe the motion of a spacecraft in the Planar Bicircular Restricted Four-Body Problem (PBRFBP).

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}) = \begin{bmatrix} v_x \\ v_y \\ 2v_y + \frac{\partial \Omega_4}{\partial x} \\ -2v_x + \frac{\partial \Omega_4}{\partial y} \end{bmatrix}$$

where

$$\Omega_4(x, y, t) = \frac{1}{2}(x^2 + y^2) + \frac{1-\mu}{r_1} + \frac{\mu}{r_2} + \frac{1}{2}\mu(1-\mu) + \frac{m_s}{r_3(t)} - \frac{m_s}{\rho^2}(x\cos(\omega_s t) + y\sin(\omega_s t)),$$

with

$$r_1 = \sqrt{(x+\mu)^2 + y^2}$$
$$r_2 = \sqrt{(x+\mu-1)^2 + y^2}$$
$$r_3(t) = \sqrt{(x - \rho\cos(\omega_s t))^2 + (y - \rho\sin(\omega_s t))^2}$$

E Jacobian Matrix of PBRFBP

Here the jacobian $A(t)$ of the Planar Bicircular Restricted Four-Body Problem (PBRFBP) right-hand-side is reported.

$$A(t) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{0}_{2\times 2} & \mathbf{I}_{2\times 2} \\ \begin{bmatrix} \frac{\partial^2 \Omega}{\partial x^2} & \frac{\partial^2 \Omega}{\partial x \partial y} \\ \frac{\partial^2 \Omega}{\partial y \partial x} & \frac{\partial^2 \Omega}{\partial y^2} \end{bmatrix} & \begin{bmatrix} 0 & 2 \\ -2 & 0 \end{bmatrix} \end{bmatrix}$$

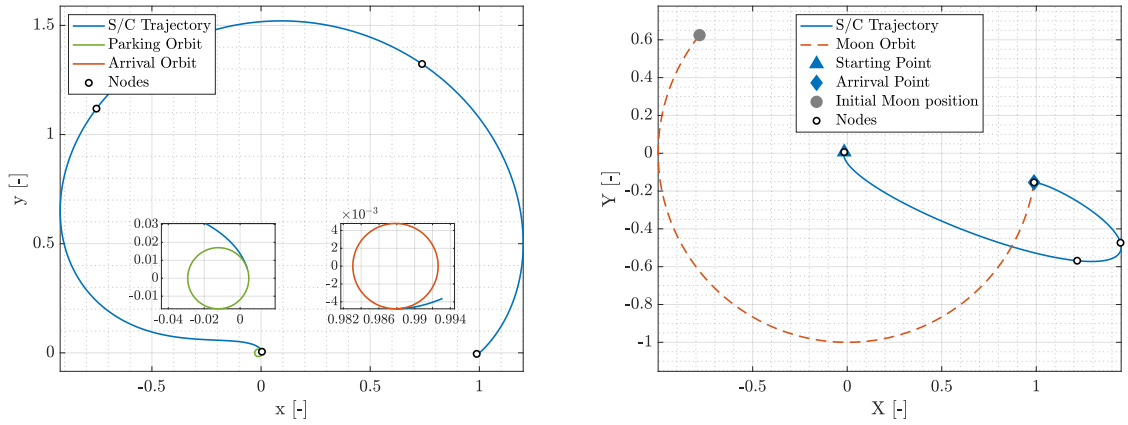
$$\begin{aligned}
\frac{\partial^2 \Omega}{\partial x^2} &= 1 - \frac{(1-\mu)}{r_1^3} + \frac{3(1-\mu)(x+\mu)^2}{r_1^5} - \frac{\mu}{r_2^3} + \frac{3\mu(x+\mu-1)^2}{r_2^5} - \frac{m_s}{r_3^3} + \frac{3m_s(x-\rho\cos(\omega_s t))^2}{r_3^5}, \\
\frac{\partial^2 \Omega}{\partial y^2} &= 1 - \frac{(1-\mu)}{r_1^3} + \frac{3(1-\mu)y^2}{r_1^5} - \frac{\mu}{r_2^3} + \frac{3\mu y^2}{r_2^5} - \frac{m_s}{r_3^3} + \frac{3m_s(y-\rho\sin(\omega_s t))^2}{r_3^5}, \\
\frac{\partial^2 \Omega}{\partial x \partial y} &= \frac{\partial^2 \Omega}{\partial y \partial x} = \frac{3(1-\mu)(x+\mu)y}{r_1^5} - \frac{3\mu(x+\mu-1)y}{r_2^5} + \frac{3m_s(x-\rho\cos(\omega_s t))(y-\rho\sin(\omega_s t))}{r_3^5}.
\end{aligned}$$

F Max Iterations

Here it is presented the multiple shooting solution obtained imposing a `ConstraintTolerance` of 10^{-10} without increasing the `MaxFunctionEvaluations` and `maxIterations` options of `fsolve`. As shown in Table 19 the solver stopped because it reached the default `maxIterations` without satisfying the requested `ConstraintTolerance`.

Table 19: Multiple Shooting performance

Parameter	**SS-FD**
Δv_{tot} [km/s]	4.0954
Δt [days]	15.9207
Iter	400
FunEval	1329
Max Constraint	3.1890e-7



Earth-Moon rotating frame | Earth centred inertial frame

Figure 18: Multiple Shooting trajectory

From Figure 18 it can be noted that the shape of trajectory is similar to the one obtained with the simple shooting. Also the results in terms of Δv_{tot} and Δt are comparable with the Simple shooting results. The obtained results in terms of initial condition, t_i and t_f are presented in Table 20.

Table 20: Multiple shooting solution in the Earth-Moon rotating frame.

$r_{x,0}$ $[DU]$	$r_{y,0}$ $[DU]$	$v_{x,0}$ $[VU]$	$v_{y,0}$ $[VU]$	t_i $[TU]$	t_f $[TU]$
0.004066	0.005190	-3.259816	10.185510	2.467	6.133

References

[1] Francesco Topputo. "On optimal two-impulse Earth–Moon transfers in a four-body model". In: Celestial Mechanics and Dynamical Astronomy 117 (2013), pp. 279–313. DOI: 10.1007/s10569-013-9513-8.