SGN – Assignment #1

Luca Frassinella, 244643

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

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

where $\Omega(x, y, z) = \dfrac{1}{2}(x^2 + y^2) + \dfrac{1-\mu}{r_1} + \dfrac{\mu}{r_2} + \dfrac{1}{2}\mu(1-\mu)$ 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 Lagrangian Points Coordinates

The coordinates of the five Lagrange points of the Earth-Moon system have been computed starting from the definition of the scalar potential for the CRTBP. The condition for all the libration points $\mathrm{L_i}$ is reported in Eq. (1):

$$\frac{\partial\Omega}{\partial x} = \frac{\partial\Omega}{\partial y} = \frac{\partial\Omega}{\partial z} = 0 \tag{1}$$

The three collinear points ($\mathrm{L_1}$, $\mathrm{L_2}$, $\mathrm{L_3}$) can be found by imposing $y = 0$ and $z = 0$. In this way, only the partial derivative of Ω in x must be investigated.
For the triangular points ($\mathrm{L_4}$, $\mathrm{L_5}$), $z = 0$, therefore $\dfrac{\partial\Omega}{\partial x} = 0$ and $\dfrac{\partial\Omega}{\partial y} = 0$ must be satisfied simultaneously.

The numerical solution has been computed using MATLAB's `fsolve`, setting the `OptimalityTolerance` option to 10^{-12}, which guarantees convergence to the correct point with a high accuracy. For each equilibrium point, different initial guesses have been provided to the solver based on plausible values.
Fig. 1 shows the coordinates of the different equilibrium points in the Earth-Moon rotating frame.

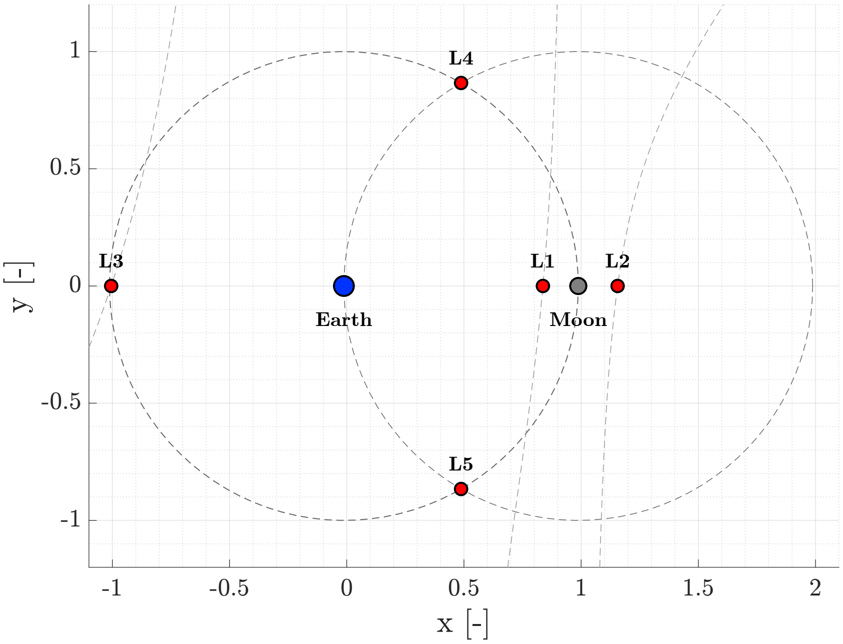

Figure 1: Earth-Moon System Lagrange Points

As can be seen in the figure, the three collinear points are located at the intersection of $\dfrac{\partial\Omega}{\partial x}$ with the x axis, while the triangular points are at the intersection between the unitary circumferences centered at $\mathrm{P_1}$ (Earth) and $\mathrm{P_2}$ (Moon), as supported by the theory.

Once the coordinates are computed, it is possible to retrieve the Jacobi Constant C associated to each of the five libration points, using the following:

$$C = 2\Omega - v^2 \tag{2}$$

where the velocity norm (v) is null as, by definition, the equilibrium points are not moving in the Earth-Moon rotating frame.

Table 1 exhibits the obtained coordinates for the five equilibrium points, as well as their associated Jacobi Constant.

	L_1	L_2	L_3	L_4	L_5
x	0.8369180073	1.1556799131	−1.0050624018	0.4878500000	0.4878500000
y	0	0	0	0.8660254038	−0.8660254038
C	3.2003380950	3.1841582164	3.0241489429	3.0000000000	3.0000000000

Table 1: Lagrangian points coordinates and Jacobi constants

1.2 Periodic Halo Orbit with C = 3.09

The aim of this section is to correct the provided initial state in order to obtain a periodic Halo orbit with an associated Jacobi Constant of 3.09.
The orbit is propagated from the initial state using an event function that stops the propagation when the condition $y = 0$ is reached. The equations of motion for the propagation are the ones of the Circular Restricted Three Body Problem (CRTBP).

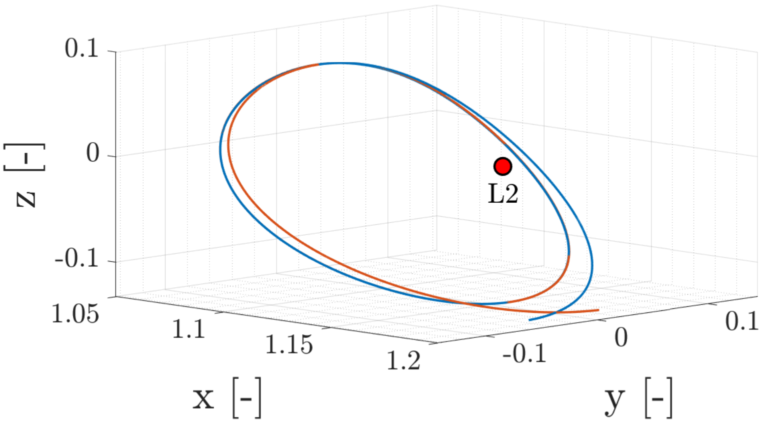

Figure 2: Propagated Trajectory from initial guess (@EMB Earth-Moon rotating frame)

As shown in Fig. 2, the initial guess does not provide a periodic Halo orbit, nor the requested Jacobi Constant. For these reasons, an iterative correction scheme must be implemented to correct the initial state. This ensures that the final trajectory satisfies both the symmetry and energy constraints required for a periodic solution.

A theoretical framework has been developed based on the first order expansion of the flow with respect to the initial state and the final time, integrating the control on the periodicity of the orbit and the value of the Jacobi Constant with an augmented linear system and an iterative correction scheme. The Taylor expansion of the flow is presented in Eq. (3).

$$\varphi(\mathbf{x_0}+\delta\mathbf{x_0}, t+\delta t) = \varphi(\mathbf{x_0}, t) + \frac{\partial\varphi(\mathbf{x_0}, t_f)}{\partial\mathbf{x_0}}\delta\mathbf{x} + \frac{\partial\varphi(\mathbf{x_0}, t_f)}{\partial t}\delta t \tag{3}$$

where $\dfrac{\delta\varphi}{\delta\mathbf{x_0}}(\mathbf{x_0}, t_f)$ is the state transition matrix at the final time $\mathbf{\Phi}(t_f)$ and $\dfrac{\delta\varphi}{\delta t}(\mathbf{x_0}, t_f)$ is the derivative of the state at the final time $\dot{\mathbf{x}}(t_f)$. The STM has been computed using a variational approach, as shown in Eq. (4). This method has been preferred with respect to a finite difference approximation because it leads to a more accurate STM computation:

$$\begin{cases} \dot{\mathbf{\Phi}} = \dfrac{\partial\mathbf{f}}{\partial\mathbf{x}}(t)\mathbf{\Phi} \\ \mathbf{\Phi}_0 = \mathbf{I}_{6x6} \end{cases} \tag{4}$$

with $\mathbf{f}$ being the right-hand side of the equations of motion for the CRTBP dynamics.
Also the Jacobi Constant has been linearized around the initial state:

$$J(\mathbf{x}_0+\delta\mathbf{x}_0) = J(\mathbf{x}_0) + \frac{\partial J(\mathbf{x}_0)}{\partial\mathbf{x}_0}\delta\mathbf{x}_0 \tag{5}$$

In Eq. (3) and Eq. (5), the left-hand side represent the desired state, that is when $y = 0$, $v_x = 0$ and $v_z = 0$ to have a perpendicular passage at the xz-plane, and $C = 3.09$ to have the desired Jacobi Constant. The first terms of the right-hand sides represent the actual state obtained by propagating the initial state $\mathbf{x}_0$. To determine the required corrections, an augmented 7x7 linear system has been built (Eq. (6)). The LHS is the vector of errors $\mathbf{e}$, while the RHS is composed by the STM matrix at the final time, the derivative of the state (7th column) and the gradient of the Jacobi Constant (7th row). The augmented matrix $\mathbf{A}$ is multiplied by the vector of corrections $\boldsymbol{\delta}$, which includes $\delta\mathbf{x}_0$, appended with the time correction δt_f.

$$\mathbf{e} = \mathbf{A}\boldsymbol{\delta}, \quad \mathbf{A} = \begin{bmatrix} \boldsymbol{\Phi} & \dot{\mathbf{x}}_f \\ \nabla J_{\mathbf{x}}(\mathbf{x}_0) & 0 \end{bmatrix}_{7\text{x}7} \tag{6}$$

Considering that the only constrained components are y, v_x, v_z, C it is possible to simplify the system by considering only the corresponding rows of $\mathbf{A}$. Furthermore, the initial state must be corrected only in terms of x_0, z_0, v_{y_0} and t_f, so only the colums of $\mathbf{A}$ related to these variables need to be used. This reduces the system to a 4x4 dimension, simplifying the computations. The final equation used to iteratively update the corrected values of the initial state is:

$$\begin{bmatrix} x_0 \\ z_0 \\ v_{y,0} \\ t_f \end{bmatrix}_{k+1} = \begin{bmatrix} x_0 \\ z_0 \\ v_{y,0} \\ t_f \end{bmatrix}_{k} - \begin{bmatrix} \boldsymbol{\Phi}_{21} & \boldsymbol{\Phi}_{23} & \boldsymbol{\Phi}_{25} & \dot{y}_f \\ \boldsymbol{\Phi}_{41} & \boldsymbol{\Phi}_{43} & \boldsymbol{\Phi}_{45} & \dot{v}_{x_f} \\ \boldsymbol{\Phi}_{61} & \boldsymbol{\Phi}_{63} & \boldsymbol{\Phi}_{65} & \dot{v}_{z_f} \\ \partial J/\partial x & \partial J/\partial z & \partial J/\partial v_y & 0 \end{bmatrix}^{-1} \begin{bmatrix} y_f - 0 \\ v_{x_f} - 0 \\ v_{z_f} - 0 \\ C_f - 3.09 \end{bmatrix} \tag{7}$$

The reduced linear system was solved iteratively using a pseudo-Newton method, updating the initial state and final time until the error in all controlled components (y, v_x, v_z, and C) falls below a predefined threshold of 10^{-12}.

The corrected initial state is showcased in Table 2, while Fig. 3 shows the obtained periodic Halo Orbit.

x	y	z	v_x	v_x	v_x
1.0590402077	0	0.0739277378	0	0.3469245709	0

Table 2: Corrected initial state of the halo orbit with $C = 3.09$

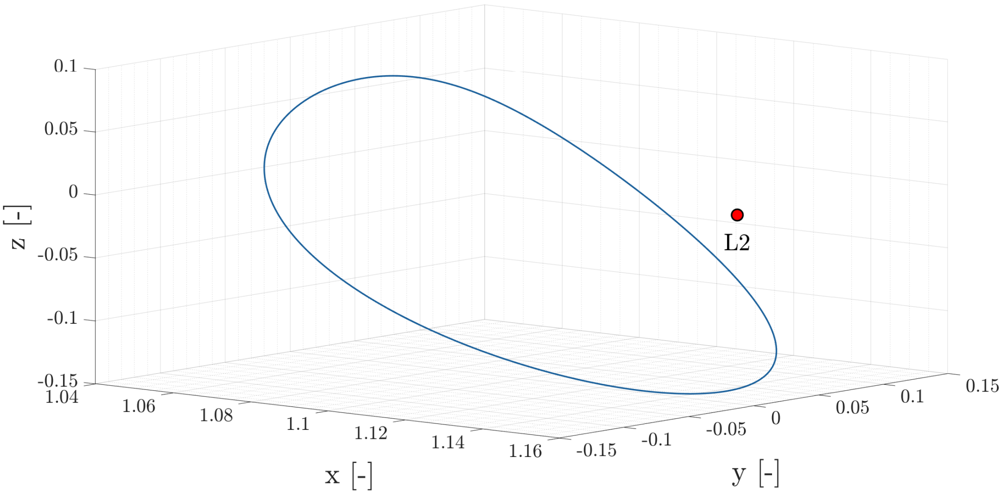

Figure 3: Periodic Halo Orbit (C=3.09) (@EMB Earth-Moon rotating frame)

1.3 Periodic Halo Orbit with C = 3.04

This section aims at finding the correction to the initial state in order to have a periodic Halo orbit with a Jacobi Constant of 3.04. To guarantee a fast convergence of the algorithm, numerical continuation is exploited. In particular, a family of 20 Halo orbits has been generated, with levels of Jacobi Constant equally spaced between 3.09 and 3.04. Starting from the initial guess provided in Section 1.2, the problem of finding the periodic orbit with the corresponding C is solved iteratively, each time updating the new initial guess as the previous step's solution, until the final Halo orbit is computed. Table 3 showcases the correct initial state.

x	y	z	v_x	v_y	v_x
1.0125655235	0	0.0672339583	0	0.5103251959	0

Table 3: Corrected initial state of the halo orbit with $C = 3.04$

Fig. 4 shows the obtained family of Halo orbits, with levels of C ranging from 3.09 to 3.04. As showcased, if the Jacobi Constant decreases the related mechanical energy increases. Therefore, the dimension of the orbit increases.

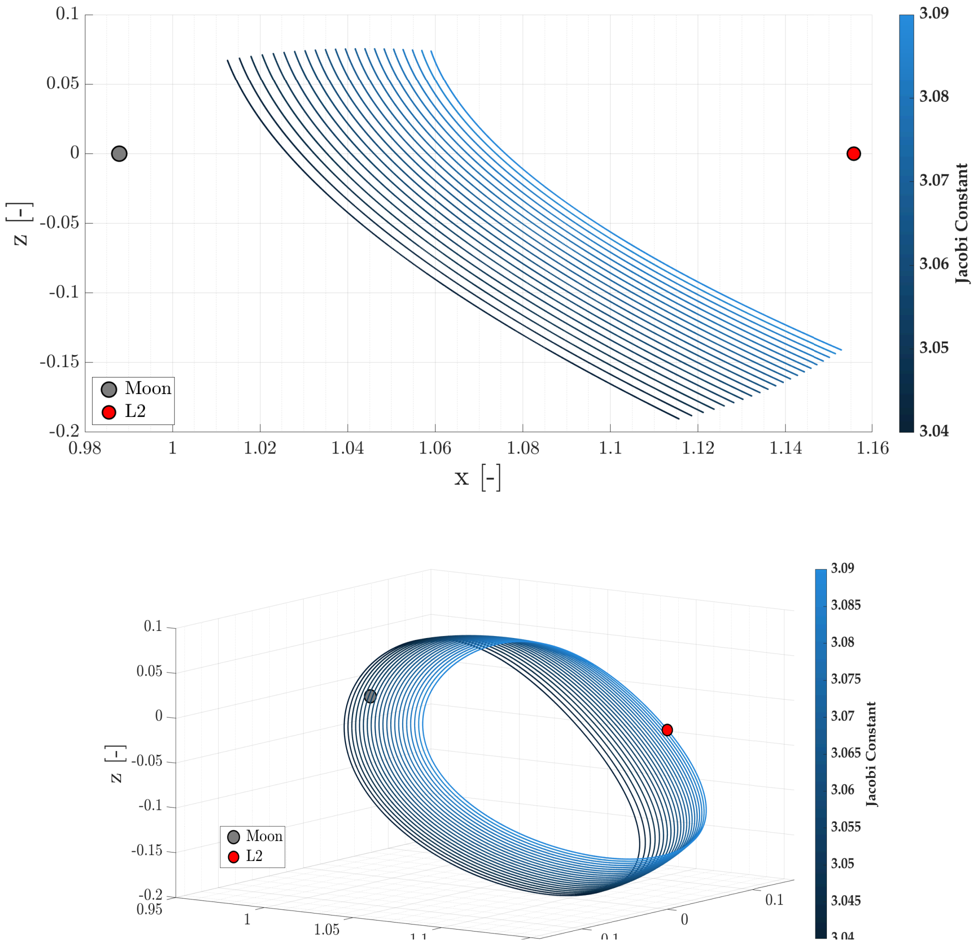

Figure 4: Family of Halo orbits with C ranging from 3.09 to 3.04 (@EMB Earth-Moon rotating frame)

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 4 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 4: 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 First Guess Solution

The aim of this section is to produce a first guess solution given $\{\alpha, \beta, t_i, \delta\}$. The adimensionalized radius r_i of the initial parking orbit has been computed using DU, starting from Earth's radius and the provided h_i (the same has been done to compute the final parking orbit r_f). Let $r_0 = r_i$ and $v_0 = \beta\sqrt{(1-\mu)/r_0}$, the initial state $\mathbf{x_0}(\alpha, \beta) = \{x_0, y_0, \dot{x}_0, \dot{y}_0\}$ is computed as:

$$x_0 = r_0\text{cos}\alpha - \mu, \quad y_0 = r_0\text{sin}\alpha, \quad \dot{x}_0 = -(v_0 - r_0)\text{sin}\alpha, \quad \dot{y}_0 = (v_0 - r_0)\text{cos}\alpha \tag{8}$$

The initial adimensionalized state is reported in Table 5:

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

Table 5: Initial guess in Earth-Moon rotating frame.

The initial guess is then propagated integrating the PBRFBP dynamics (Section 2.2, Topputo, 2013)*. Fig. 5 shows the trajectory in the Earth-Moon rotating frame. As can be seen, the first guess solution does not yield an optimal nor acceptable transfer, as the final propagated state does not reach the Moon parking orbit. The trajectory has then been rotated from the rotating frame to the Earth Centered Inertial Frame (ECI), with the methodology outlined in Appendix A.

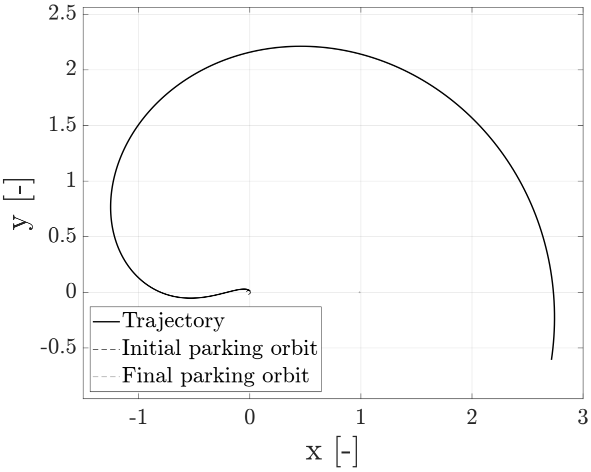

Figure 5: Initial Guess Trajectory (@EMB Earth-Moon rotating frame)

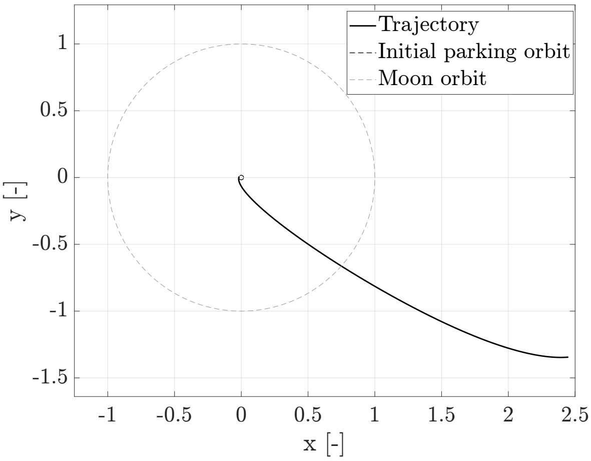

Figure 6: Initial Guess Trajectory (@Earth ECI)

2.2 Simple Shooting

In this section, the two-impulse Earth-Moon transfer is optimized using simple shooting techniques, starting from the initial guess computed in Section 2.1 and using $\{\mathbf{x}_i, t_i, t_f\}$ as optimization variables. A brief recap of the problem statement is reported hereafter.

Let $\mathbf{x}_i$ be the initial state at the Earth circular parking orbit (of adimensionalized radius r_i), t_i and t_f the initial and final adimensionalized propagation times, respectively.
Let $\mathbf{x}_f = \varphi(\mathbf{x}_i, t_i; t_f)$ be the propagated final state using the PBRFBP dynamic model. The equality constraints to be satisfied by the Earth-Moon transfer are:

$$\boldsymbol{\psi}_i(\mathbf{x}_i) = \begin{cases} (x_i+\mu)^2 + y_i^2 - r_i^2 & = 0 \\ (x_i+\mu)(\dot{x}_i - y_i) + y_i(\dot{y}_i + x_i + \mu) & = 0 \end{cases} \tag{9}$$

$$\boldsymbol{\psi}_f(\mathbf{x}_f) = \begin{cases} (x_f + \mu - 1)^2 + y_f^2 - r_f^2 & = 0 \\ (x_f + \mu - 1)(\dot{x}_f - y_f) + y_f(\dot{y}_f + x_f + \mu - 1) & = 0 \end{cases} \tag{10}$$

The total cost of the transfer is:

$$\Delta v(\mathbf{x}_i, t_i, t_f) = \Delta v_i(\mathbf{x}_i) + \Delta v_f(\varphi(\mathbf{x}_i, t_i; t_f)) \tag{11}$$

where:

$$\Delta v_i = \sqrt{(\dot{x}_i - y_i)^2 + (\dot{y}_i + x_i + \mu)^2} - \sqrt{\frac{1-\mu}{r_i}} \tag{12}$$

$$\Delta v_f = \sqrt{(\dot{x}_f - y_f)^2 + (\dot{y}_f + x_f + \mu - 1)^2} - \sqrt{\frac{\mu}{r_f}} \tag{13}$$

The optimization problem consists of finding $\{\mathbf{x}_i, t_i, t_f\}$, such that $\boldsymbol{\psi}_i(\mathbf{x}_i) = \mathbf{0}$, $\boldsymbol{\psi}_f(\varphi(\mathbf{x}_i, t_i; t_f)) = \mathbf{0}$ and the cost function $\Delta v(\mathbf{x}_i, t_i, t_f)$ is minimized.
This problem is mapped to an algebraic nonlinear programming (NLP) problem, stated as:

$$\min_{(\mathbf{x}_i, t_i, t_f)} \quad f(\mathbf{x}_i, t_i, t_f) \quad \text{s.t.} \quad \mathbf{c}(\mathbf{x}_i, t_i, t_f) = \mathbf{0} \quad \text{where :} \quad \mathbf{c} = \begin{cases} \boldsymbol{\psi}_i(\mathbf{x}_i) \\ \boldsymbol{\psi}_f(\varphi(\mathbf{x}_i, t_i; t_f)) \end{cases} \tag{14}$$

The NLP problem has been solved using MATLAB's `fmincon` function, using `active-set` as algorithm and setting both `ConstraintTolerance` and `OptimalityTolerance` to 10^{-10}, in order to obtain precise results.

The first optimization was performed without providing the analytical gradients of the objective function and equality constraints to the solver. Fig. 7 shows the obtained trajectory in the rotating frame.

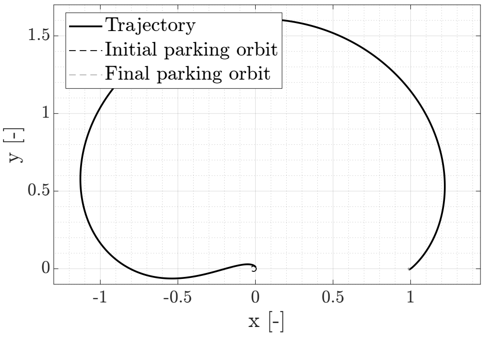

Figure 7: Optimal simple shooting trajectory (no gradients) (@EMB Earth-Moon Rotating Frame)

The results of Table 6 show the optimal variables for the Earth-Moon transfer, expressed in the Earth-Moon rotating frame.

As requested, a second optimization was performed by providing the analytical derivatives for the objective function and the equality constraints, integrating the state transition matrix with a variational approach (refer to Appendix B for all the derivation).
As expected, this method improved the convergence of `fmincon`, decreasing the number of function evaluations of each iteration. Furthermore, an improvement is observed also in the optimal cost of the transfer, which is decreased from $\Delta v = 4.0076$ of the first optimization to $\Delta v = 4.0068$.
The optimal variables for the second optimization are reported in Table 6, expressed with 6-digit accuracy to facilitate the comparison between the two methods.

Gradients	$r_{x,0}$ $[DU]$	$r_{y,0}$ $[DU]$	$v_{x,0}$ $[VU]$	$v_{y,0}$ $[VU]$	t_i $[TU]$	t_f $[TU]$
False	0.001610	0.010027	-6.299679	8.645478	2.0264	6.0331
True	0.001620	0.010013	-6.290741	8.651414	2.1982	6.2032

Table 6: Simple shooting solutions in the Earth-Moon rotating frame.

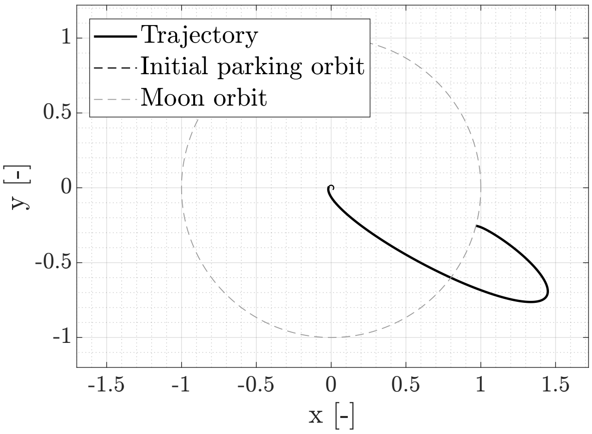

Figure 8: Optimal simple shooting trajectory (no gradients) (@Earth ECI)

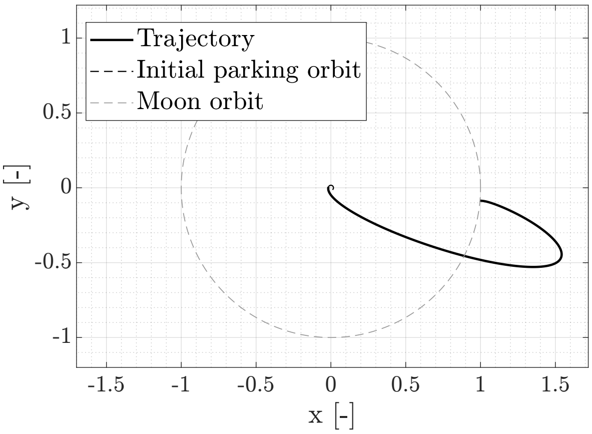

Figure 9: Optimal simple shooting trajectory (with gradients) (@Earth ECI)

Gradients	**Error**	**Value**	**Units**
False	$\|\|\mathbf{r}(t_f) - \mathbf{r}_f\|\|$	$4.6121e-06$	m
	$\|\|\mathbf{v}(t_f) - \mathbf{v}_f\|\|$	$6.3474e-09$	m/s
True	$\|\|\mathbf{r}(t_f) - \mathbf{r}_f\|\|$	$4.0686e-06$	m
	$\|\|\mathbf{v}(t_f) - \mathbf{v}_f\|\|$	$6.3102e-09$	m/s

Table 7: Dimensionalized errors at Moon arrival orbit (Simple Shooting)

Fig. 8 and Fig. 9 show the obtained trajectories in the ECI frame for the two optimizations. In Table 7 the dimensionalized errors at the reference arrival orbit are provided for both simple shooting methods:

It can be seen that, due to the low tolerance provided for the constraints, the discrepancy in the final position and velocity is in the order of $10^{-6}\,m$ and $10^{-9}\,m/s$. Furthermore, by providing the analytical derivatives to `fmincon`, the obtained solution yields slightly more precise results, as confirmed by the lower final constraint residuals.

Fig. 10 and Fig. 11 offer a visual representation of the departure from Earth's parking orbit and the arrival to the Moon's circular orbit.

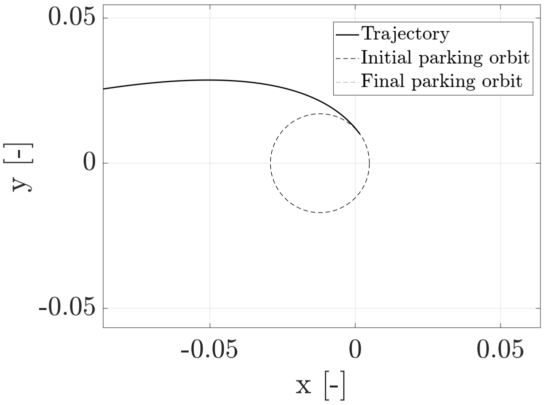

Figure 10: Earth Departure

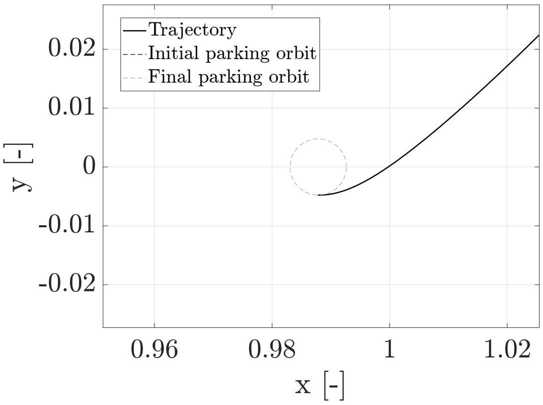

Figure 11: Moon Arrival

2.3 Multiple Shooting

In this section, a multiple shooting approach with $N = 4$ is implemented to optimize the Earth-Moon transfer.
The NLP variable vector is therefore defined as:

$$\mathbf{y} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, t_1, t_4\} \tag{15}$$

This approach uses a discretized uniform time grid from t_i to t_f, made of $N = 4$ points such that $t_1 = t_i$ and $t_4 = t_f$. Let $\mathbf{x}_j = \mathbf{x}(t_j)$ be the solutions of the PBRFBP dynamics at the j-th mesh point ($j = 1, ..., N$). The multiple shooting approach consists in integrating the dynamics within the $N - 1$ intervals ($[t_j, t_{j+1}]$), taking as initial condition $\mathbf{x}_j$. In this case, the vector of equality constraints is augmented with respect to the simple shooting case:

$$\mathbf{c}(\mathbf{y}) = \{\boldsymbol{\psi}_1, \boldsymbol{\zeta}_1, ..., \boldsymbol{\zeta}_{N-1}, \boldsymbol{\psi}_N\} \tag{16}$$

where $\boldsymbol{\zeta}_j$ represents the defects between the propagated state starting from $\mathbf{x}_j$ and the state at the following node (Eq. (17)). These vectors must vanish in order to achieve the continuity of position and velocity throughout the entire transfer.

$$\boldsymbol{\zeta}_j = \varphi(\mathbf{x}_j, t_j; t_{j+1}) - \mathbf{x}_{j+1}, \quad j = 1, ..., N - 1 \tag{17}$$

Moreover, the trajectory optimization is further constrained by the fact that each of the four states must be such that an impact with either the Earth or the Moon is avoided. Furthermore, even the initial and final times are constrained because the transfer duration must be strictly positive (i.e. $t_1 - t_N < 0$). The vector of nonlinear inequality constraints is:

$$\mathbf{g}(\mathbf{y}) = \{\boldsymbol{\eta}_1, ..., \boldsymbol{\eta}_N, \tau\} \tag{18}$$

The NLP formulation of the multiple shooting problem, with $4N + 2$ variables and $4N$ constraints is:

$$\min_{\mathbf{y}} \quad f(\mathbf{y}) \quad \text{s.t.} \quad \begin{cases} \mathbf{c}(\mathbf{y}) = \mathbf{0} \\ g_j(\mathbf{y}) < 0, \quad j = 1, ...2N + 1 \end{cases} \tag{19}$$

To speed up the convergence of the algorithm, analytic gradients for both the objective function and the equality constraints were provided to the `fmincon` solver (refer to Appendix C for the complete derivation). The setting for the `ConstraintTolerance` was increased to 10^{-7}: this choice stems from the necessity to find a balance between a fast-converging solution and an accurate one. As can be seen in Table 8, the residuals at the final node ensure a very high precision of the solution. However, the internal nodes the continuity is less precise, with errors as large as a few tens of meters. Better solutions could be found by further tightening the tolerances for the internal constraints.

Error	Value	Units
$\|\|\mathbf{r}(t_f) - \mathbf{r}_f\|\|$	$6.2143e - 08$	m
$\|\|\mathbf{v}(t_f) - \mathbf{v}_f\|\|$	$7.3349e - 12$	m/s

Table 8: Final state error at Moon orbit arrival (Multiple Shooting).

In Table 9 the optimal solution in terms of initial state and transfer times is reported.

However, it is important to note that the errors at the final nodes, as discussed in Section 2.2 and Section 2.3, rely on the assumption of the PBRFBP, which assumes planar and circular orbit. Consequently, this discrepancies are expected to be higher in real-world scenarios.

The total adimensionalized cost of the transfer is $\Delta v = 3.99735$, further improving the optimization of the Simple Shooting techniques. This result was expected, as it confirms the

$r_{x,0}$ $[DU]$	$r_{y,0}$ $[DU]$	$v_{x,0}$ $[VU]$	$v_{y,0}$ $[VU]$	t_i $[TU]$	t_f $[TU]$
0.004066	0.005191	−3.260185	10.185396	2.4657	6.1320

Table 9: Multiple shooting solution in the Earth-Moon rotating frame.

better precision of multiple shooting techniques when dealing with nonlinear dynamics (such as the PBRFBP model) and long transfer times.

Fig. 12 and Fig. 13 represent the optimal transfer in the Earth-Moon rotating frame and in the Earth Centered Inertial frame, respectively.

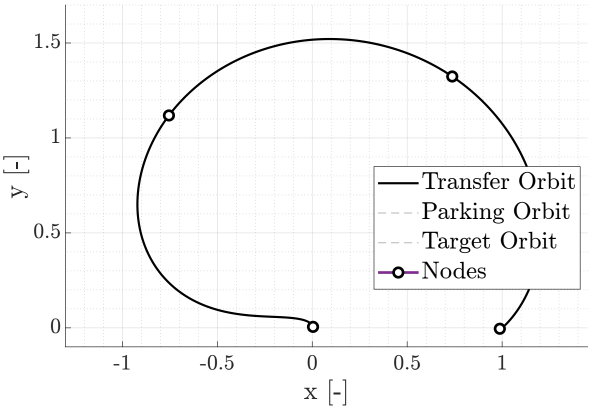

Figure 12: Multiple Shooting Optimal Trajectory (@EMB Earth-Moon Rotating Frame)

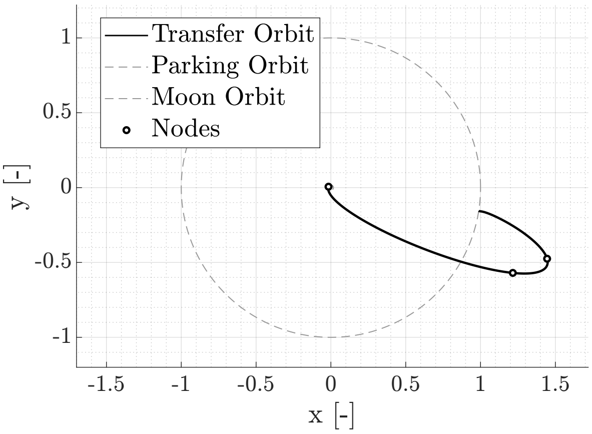

Figure 13: Multiple Shooting Optimal Trajectory (@Earth ECI)

2.4 N-body Propagation

The aim of this section is to validate the optimal trajectory computed in Section 2.2, by comparing it to an N-body propagator, which offers a more realistic representation of the spacecraft's dynamics, accounting for the gravitational influence of all the main celestial bodies of the Solar System and a 3D representation of the trajectory, unlike the PBRFBP model.

Since the N-body propagation needs an initial epoch, a zero-finding problem has been implemented to search for a time instant in which the relative position of the Sun and the Moon is the same as of the optimization problem of Section 2.2.

Considering the definition of the angle $\theta(t)$ as the angle between the rotating frame's x-axis, aligned with the position vector from the Earth-Moon System Barycenter (EMB) to the Moon, and the Sun direction, the target value θ_{target} at epoch t_i is computed as $\theta_{\text{target}} = \omega_s t_i$, where ω_s is the scaled angular velocity of the Sun. A zero-finding problem is then formulated, with the solution interval corresponding to one Moon revolution starting from the epoch t_0 (2024 Sep 28 00:00:00.000 TDB). Using MATLAB's `fzero`, the epoch where $\theta(t) = \theta_{\text{target}}$ is found. Adding the time of flight, the final epoch is computed. The results are showcased in Table 10, along with the initial state vector in the ECI frame.

Symbol	**Calendar epoch** (UTC)
t_i	2024-10-12T11:40:24.970
t_f	2024-10-29T21:05:11.122

$r_{x,0}$ $[km]$	$r_{y,0}$ $[km]$	$r_{z,0}$ $[km]$	$v_{x,0}$ $[km/s]$	$v_{y,0}$ $[km/s]$	$v_{z,0}$ $[km/s]$
−6223.6306	2026.1379	0.0	−3.39799	−10.43750	0.0

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

Once the initial epoch is defined, the dynamics of the N-body model are implemented. The equations of motion account for the central body's gravitational acceleration $\mathbf{a}_0$ and the contributions $\mathbf{a}_i$ from additional celestial bodies. Hence, the equations of motion are:

$$\frac{d}{dt}\begin{pmatrix}\mathbf{r}\\ \mathbf{v}\end{pmatrix} = \begin{pmatrix}\mathbf{v}\\ \mathbf{a}\end{pmatrix}, \quad \mathbf{a} = \mathbf{a}_0 + \sum_{i\neq 0}\mathbf{a}_i, \quad \text{where :}\mathbf{a}_i = -\frac{\mu_i}{|\mathbf{r}-\boldsymbol{\rho}_i^3|}(\mathbf{r}+\boldsymbol{\rho}_i f(q)) \tag{20}$$

where μ_i is the gravitational parameter of the i-th body, ρ_i the distance from the central body, and $\mathbf{r}_i$ the position vector. The perturbation function $f(q)$ is defined as:

$$f(q) = \frac{q(3+3q+q^2)}{1+(1+q)^{3/2}}, \quad \text{with :} \quad q = \frac{\mathbf{r}\cdot(\mathbf{r}-2\boldsymbol{\rho}_i)}{|\boldsymbol{\rho}_i|^2} \tag{21}$$

After numerically integrating these equations, the trajectory is compared with the one from Section 2.2.

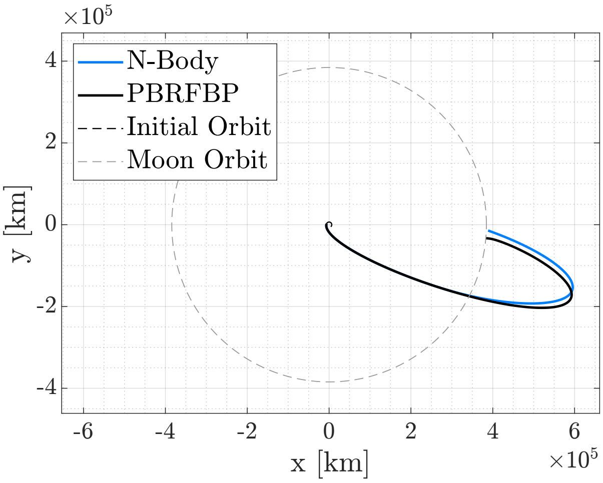

Figure 14: Comparison between N-body propagation and PBRFBP propagation (@Earth ECI)

As shown in Fig. 14, the N-body trajectory differs slightly from the PBRFBP trajectory, particularly in its out-of-plane components, which result from the three-dimensional nature of the N-body dynamics. The PBRFBP trajectory, optimized for the planar model, perfectly matches the Moon, unlike the N-body propagated trajectory. This difference is due to the fact that the initial state used for the propagation was optimized for the PBRFBP dynamics.
Moreover, the PBRFBP trajectory is fully contained in the Moon's orbital plane, and does not take into account the spatial deviations present in the N-body model. This difference is due to the 3D nature of the N-body dynamics.

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	km^3/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 aligned with the angular momentum, while the W-axis is pointing inwards, i.e., towards the Earth.

3.1 Debris Spatial Density

The evolution of the debris spatial density as a function of orbital altitude is showcased in Section 3.1.

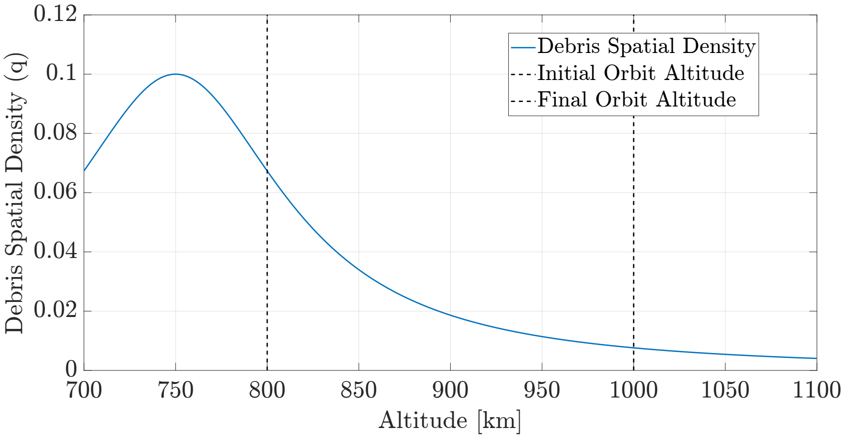

Figure 15: Debris Spatial Density evolution over altitude range of interest

As can be seen, the peak in debris density is located 50 km below the altitude of the initial orbit, while for the range of heights of interest from the initial and final orbit the population of resident space object shows a decreasing trend.

The initial and target state were computed as shown in , by exploiting the circular geometry of the orbits and the knowledge on the final orbit inclination.

$$\mathbf{x}_i = \begin{pmatrix} R_e + h_i \\ 0 \\ 0 \\ 0 \\ \sqrt{\dfrac{\mu}{Re + h_i}} \\ 0 \end{pmatrix} \quad \mathbf{x}_f = \begin{pmatrix} R_e + h_f \\ 0 \\ 0 \\ 0 \\ \sqrt{\dfrac{\mu}{Re + h_f}}\cos\Delta i \\ \sqrt{\dfrac{\mu}{Re + h_f}}\sin\Delta i \end{pmatrix} \tag{22}$$

The obtained results are shown in Table 12:

$r_{x,i}$ $[km]$	$r_{y,i}$ $[km]$	$r_{z,i}$ $[km]$	$v_{x,i}$ $[km/s]$	$v_{y,i}$ $[km/s]$	$v_{z,i}$ $[km/s]$
7178.136600	0.0	0.0	0.0	7.45183148	0.0

$r_{x,f}$ $[km]$	$r_{y,f}$ $[km]$	$r_{z,f}$ $[km]$	$v_{x,f}$ $[km/s]$	$v_{y,f}$ $[km/s]$	$v_{z,f}$ $[km/s]$
7178.136600	0.0	0.0	0.0	7.34950906	0.09621034

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

3.2 Adimensionalized parameters

Using as reference the provided DU and MU, the time and velocity units were computed from:

$$TU = \sqrt{\frac{DU^3}{\mu}}, \quad VU = \frac{DU}{TU} \tag{23}$$

Table 13 reports all the adimensionalized parameters obtained using the reference quantities. Refer to Appendix D for the complete derivation of the adimensionalized parameters.

μ [-]	ρ_0 [-]	m_0 [-]	T_{max} [-]	I_{sp} [-]	g_0 [-]	DU [km]	MU [kg]	TU [s]	VU [km/s]
1	0.99303	1	3.8780e-04	3.23896	1.2681	7178.1366	1000	963.27	7.45183

Table 13: Adimensionalized Parameters and Constants

In Eq. (24), the adimensionalized initial and target states are reported:

$$\mathbf{x}_i = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \qquad \mathbf{x}_f = \begin{pmatrix} 1.02786 \\ 0 \\ 0 \\ 0 \\ 0.98627 \\ 0.01291 \end{pmatrix} \tag{24}$$

3.3 PMP Formulation

In this section, the problem is formulated based on the Pontryagin Maximum Principle. Since the objective function aims at minimizing the risk of impact with resident space debris during the orbit raising manoeuvre, the lagrangian function of the problem is:

$$l(\mathbf{x}, \mathbf{u}, t) = q(\rho(t)) \tag{25}$$

The problem is formulated as a time-optimal problem ($\mathbf{u} = 1$), because the optimal transfer will be the one minimizing the time spent in the region with high space debris density. The problem is therefore formalized as follows:

$$\min_{(\hat{\boldsymbol{\alpha}}, u) \in \Omega} \int_{t_0}^{t_f} l(\mathbf{x}, \mathbf{u}, t) dt \quad \text{s.t.} \quad \begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \hat{\boldsymbol{\alpha}}, t) \\ \mathbf{x}(t_0) = \mathbf{x}_0 \\ \mathbf{r}(t_f) = \mathbf{r}_f \\ \mathbf{v}(t_f) = \mathbf{v}_f \\ \lambda_m(t_f) = 0 \end{cases} \tag{26}$$

Where Ω is the set of admissible control actions, $\hat{\boldsymbol{\alpha}}$ is the thrust direction and u is the throttle factor:

$$\Omega = \{(u, \hat{\boldsymbol{\alpha}}) \, : u \in [0, 1], ||\hat{\boldsymbol{\alpha}}|| = 1\} \tag{27}$$

The Hamiltonian of the problem is defined as:

$$H = l + \boldsymbol{\lambda}^T \mathbf{f} = q(\rho) + \begin{pmatrix} \boldsymbol{\lambda}_r \\ \boldsymbol{\lambda}_v \\ \lambda_m \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ -\dfrac{\mu}{r^3}\mathbf{r} + \dfrac{\mathbf{T_{max}}}{\mathbf{m}}\hat{\boldsymbol{\alpha}} \\ -\dfrac{T_{max}}{I_{sp} g_0} \end{pmatrix} \tag{28}$$

Applying the Pontryagin's Maximum Principle, which identifies the optimal control policy among admissible controls by minimizing the Hamiltonian, the optimal control is derived in terms of the thrust direction and control level (u^*) . The latter remains constant at 1 due to the time-optimal nature of the problem, while $\hat{\boldsymbol{\alpha}}^*$, defined in Eq. (29), ensures that the thrust direction aligns with the negative gradient of the velocity costate.

$$\hat{\boldsymbol{\alpha}}^* = -\frac{\boldsymbol{\lambda}_v}{||\boldsymbol{\lambda}_v||} \tag{29}$$

Consequently, the Euler Lagrange Equations (embedding $\hat{\boldsymbol{\alpha}}^*$ and u^*) are defined:

$$\begin{cases} \dot{\mathbf{x}} = \dfrac{\partial H}{\partial \boldsymbol{\lambda}} \\ \dot{\boldsymbol{\lambda}} = -\dfrac{\partial H}{\partial \mathbf{x}} \end{cases} = \begin{cases} \dot{\mathbf{r}} = \dot{\mathbf{v}} \\ \dot{\mathbf{v}} = -\dfrac{\mu}{r^3}\mathbf{r} - \dfrac{T_{max}}{m}\dfrac{\boldsymbol{\lambda}_v}{||\boldsymbol{\lambda}_v||} \\ \dot{m} = -\dfrac{T_{max}}{I_{sp} g_0} \\ \dot{\boldsymbol{\lambda}}_r = -\dfrac{\partial q(\rho)}{\partial \mathbf{r}} - \dfrac{3\mu}{r^5}(\mathbf{r}\cdot\boldsymbol{\lambda}_v)\mathbf{r} + \dfrac{\mu}{r^3}\boldsymbol{\lambda}_v \\ \dot{\boldsymbol{\lambda}}_v = -\boldsymbol{\lambda}_r \\ \dot{\lambda}_m = -\dfrac{\lambda_v T_{max}}{m^2} \end{cases} \tag{30}$$

Additionally, to find t_f the proper transversality condition is reported in Eq. (31). Since the target is a fixed point in space, $\dot{\boldsymbol{\psi}}(t_f) = 0$.

$$H(t_f) - \boldsymbol{\lambda}(t_f)\cdot\dot{\boldsymbol{\psi}}(t_f) = 0 \quad \longrightarrow H(t_f) = 0 \tag{31}$$

To solve the Two-Point Boundary Value Problem (TPBVP), the goal is to find $\boldsymbol{\lambda}_0$ and t_f such that the propagated final state and costates $\varphi([\mathbf{x}_0; \boldsymbol{\lambda}_0], t_0, t_f)$ produce a solution that minimizes the objective function by solving the E-L equations (30) while simultaneously satisfying the boundary conditions (26), by solving the following zero-finding problem:

$$\begin{cases} \mathbf{r}(tf) - \mathbf{r}_f = 0 \\ \mathbf{v}(tf) - \mathbf{v}_f = 0 \\ \lambda_m(t_f) = 0 \\ H(t_f) = 0 \end{cases} \tag{32}$$

3.4 Continuous Guidance Solution

This section aims at finding the solution to continuous guidance problem. The initial guess in terms of $\boldsymbol{\lambda}_0$ and t_f is generated using random number such that $\lambda_{0,i} \in [-250; +250]$, while $t_f = 20\pi$. After providing the initial guess, the zero-finding algorithm described in Appendix E is solved using MATLAB's `fsolve`, where **res** is the the vector of residuals that must vanish. The algorithm used by the solver was chosen as `levenberg-marquardt`, due to its inherent better performances when dealing with initial guess solutions that are far from the optimal one.

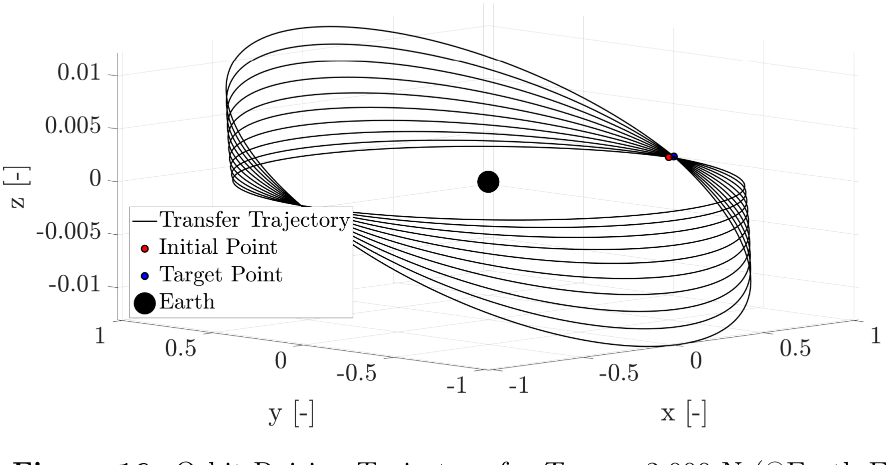

Figure 16: Orbit Raising Trajectory for $T_{max} = 3.000$ N (@Earth ECI)

To improve the efficiency and the accuracy of the convergence, the analytical derivatives of the boundary conditions and the Hamiltonian were provided to the solver. In particular, let $y = [\boldsymbol{\lambda}_0; t_f]$ be the set of variables, and $\mathbf{F}$ the zero finding function stated in Eq. 32, the Jacobian of $\mathbf{F}$ is defined as:

$$\frac{\partial \mathbf{F}}{\partial \mathbf{y}} = \begin{bmatrix} \dfrac{\partial \mathbf{r}_f}{\partial \boldsymbol{\lambda}_0} & \dfrac{\partial \mathbf{r}_f}{\partial t_f} \\ \dfrac{\partial \mathbf{v}_f}{\partial \boldsymbol{\lambda}_0} & \dfrac{\partial \mathbf{v}_f}{\partial t_f} \\ \dfrac{\partial \lambda_{m,f}}{\partial \boldsymbol{\lambda} 0} & \dfrac{\partial \lambda m, f}{\partial t_f} \\ \dfrac{\partial H_f}{\partial \boldsymbol{\lambda}_0} & 0 \end{bmatrix}_{8\times 8} \tag{33}$$

By applying the chain rule of derivatives the final term can be reformulated as shown in Eq. (34):

$$\frac{\partial H_f}{\partial \boldsymbol{\lambda}_0} = \frac{\partial H_f}{\partial \mathbf{x}_f}\frac{\partial \mathbf{x}_f}{\partial \boldsymbol{\lambda}_0} + \frac{\partial H_f}{\partial \boldsymbol{\lambda}_f}\frac{\partial \boldsymbol{\lambda}_f}{\partial \boldsymbol{\lambda}_0} = -\dot{\boldsymbol{\lambda}}_f \frac{\partial \mathbf{x}_f}{\partial \boldsymbol{\lambda}_0} + \dot{\mathbf{x}}_f \frac{\partial \boldsymbol{\lambda}_f}{\partial \boldsymbol{\lambda}_0} \tag{34}$$

Finally, being $\mathbf{f}$ the right-hand side of Eq. (30), and computing its state transition matrix $\boldsymbol{\Phi}(t_i, t_f)$ the Jacobian of $\mathbf{F}$ is formulated as:

$$\frac{\partial \mathbf{F}}{\partial \mathbf{y}} = \begin{bmatrix} \boldsymbol{\Phi}\mathbf{r}_f, \boldsymbol{\lambda}_0 & \mathbf{f}\mathbf{r}_f \\ \boldsymbol{\Phi}\mathbf{v}_f, \boldsymbol{\lambda}_0 & \mathbf{f}\mathbf{v}_f \\ \boldsymbol{\Phi}\lambda_m, \boldsymbol{\lambda}_0 & \mathbf{f}\lambda_m \\ -\mathbf{f}\boldsymbol{\lambda}_f\boldsymbol{\Phi}\mathbf{x}f, \boldsymbol{\lambda}_0 + \mathbf{f}\mathbf{x}f\boldsymbol{\Phi}\boldsymbol{\lambda}_f, \boldsymbol{\lambda}_0 & 0 \end{bmatrix}_{8\times 8} \tag{35}$$

where the subscripts indicate the rows and columns of the STM submatrix, respectively.

The optimal solution for the orbit raising transfer in terms of initial costate vector and final time is reported in Table 14. Fig. 16 shows the orbit raising trajectory from the initial to the target point.

It must be noted that, during the solution of the zero-finding problem, an alternative transfer was occasionally found, corresponding to a final time of $t_f \approx 22\pi$. However, this solution was discarded as it required an additional orbital period to complete the transfer. This trajectory, although being feasible, represents a sub-optimal solution for a time-optimal problem. For these reasons, this solution was discarded.

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

Table 14: Optimal orbit raising transfer solution ($T_{\max} = 3.000$ N).

Table 15 showcases the final state error in terms of displacements from the target state.

To assess the validity of the obtained solution, the time evolution of the Hamiltonian over the entire transfer was computed, as reported in Fig. 17. The Hamiltonian is constant, which is expected in problems that are not time-dependent and time-optimal solutions. Fig. 18 shows

Error	Value	Units
$\|\|\mathbf{r}(t_f) - \mathbf{r}_f\|\|$	$2.0375e-09$	km
$\|\|\mathbf{v}(t_f) - \mathbf{v}_f\|\|$	$2.0628e-09$	m/s

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

the time evolution of the primer vector in the NTW frame. The transformation from ECI to NTW was computed as follows:

$$\hat{\boldsymbol{\alpha}}_{NTW} = \mathbf{R} \times \hat{\boldsymbol{\alpha}}_{eci}, \quad \mathbf{R} = \left[\hat{\mathbf{N}}, \hat{\mathbf{T}}, \hat{\mathbf{W}}\right] \tag{36}$$

where:

$$\hat{\mathbf{N}} = \frac{\mathbf{r}_{ECI} \times \mathbf{v}_{ECI}}{||\mathbf{r}_{ECI} \times \mathbf{v}_{ECI}||}, \quad \hat{\mathbf{T}} = \frac{\mathbf{v}_{ECI}}{||\mathbf{v}_{ECI}||}, \quad \hat{\mathbf{W}} = \hat{\mathbf{N}} \times \hat{\mathbf{T}} \tag{37}$$

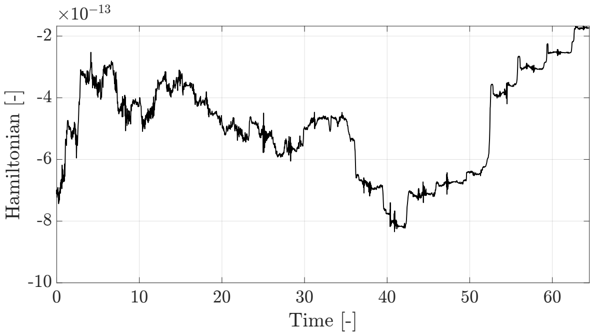

Figure 17: Time evolution of Hamiltonian ($T_{max} = 3.000$ N)

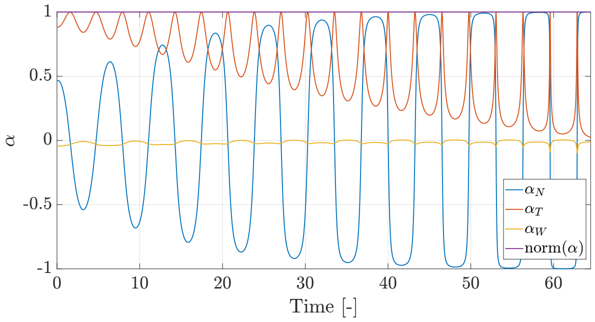

Figure 18: Time evolution of prime vector components in the NTW frame

As shown in Fig. 18, the $\hat{\text{T}}$ component of the primer vector remains consistently positive, with higher amplitudes observed at the beginning of the transfer. This suggests that the thrust is predominantly tangential in this region, aligning with the expectation of achieving a faster transfer early on when debris density is higher.
The $\hat{\text{N}}$ component exhibits notable oscillations with a period matching that of one orbit. This behavior is likely due to the out-of-plane thrust being applied periodically at the line of nodes, with positive amplitudes near the ascending node and negative ones at the descending node.
The $\hat{\text{W}}$ component stays close to zero throughout the transfer, indicating that thrust in this direction is unnecessary, as the manoeuvre focuses on modifying the orbit's semi-major axis and inclination.

3.5 Analysis with Reduced Thrust Level

In this final section, the continuous guidance problem has been solved for a lower thrust level of $T_{max} = 2.860$ N. The procedure to find the optimal initial costate vector $\boldsymbol{\lambda}_0$ and the final time t_f is the same outlined in Algorithm 1, with the exploitation of numerical continuation to improve the speed of convergence of the algorithm. A vector of equally spaced thrust levels ranging from 3.000 N to 2.860 N has been computed to iteratively solve the zero-finding problem. At each iteration, the previous step's solution was used as the initial guess. For the optimization, the chosen algorithm is `trust-region-dogleg`, because of its better performances when dealing with problems for which the initial guess is relatively close to the optimal solution, as is the case when exploiting numerical continuation. The final solution is displayed in Table 16.

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	2.2973	-11.9293	-592.7890	-920.3698	17.3814

Table 16: Optimal orbit raising transfer solution ($T_{\text{max}} = 2.860$ N).

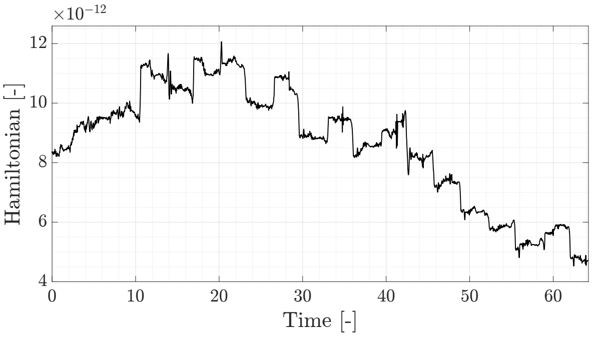

Figure 19: Time evolution of Hamiltonian ($T_{max} = 2.860$ N)

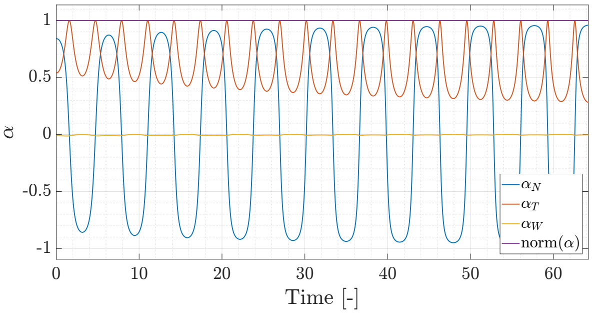

Figure 20: Time evolution of prime vector components in the NTW frame

Also in this case, the solution is validated by checking that the the Hamiltonian remains constant throughout the entire transfer, as shown in Fig. 19.

Table 17 shows the final state error with respect to the target state:

Error	**Value**	**Units**
$\|\|\mathbf{r}(t_f) - \mathbf{r}_f\|\|$	$1.0696e - 08$	km
$\|\|\mathbf{v}(t_f) - \mathbf{v}_f\|\|$	$1.0621e - 08$	m/s

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

As shown in Fig. 20, the primer vector exhibits more pronounced oscillations compared to the previous case (Fig. 18). The vector components display more consistent amplitudes, suggesting that the thrusting strategy is more efficient in the low-thrust scenario. This improved efficiency likely explains the reduced time required to reach the final target, as highlighted by the results in Table 16.

Appendix

A Rotation to ECI frame

An orbit expressed in the rotating frame, with $\mathbf{x} = (x(t), y(t), \dot{x}(t), \dot{y}(t))$, can be rotated in the ECI frame, with $\mathbf{X} = (X(t), Y(t), \dot{X}(t), \dot{Y}(t))$ using the following:

$$X(t) = (x(t) + \mu)\text{cos}(t) - y(t)\text{sin}(t) \tag{38}$$

$$Y(t) = (x(t) + \mu)\text{sin}(t) + y(t)\text{cos}(t) \tag{39}$$

$$\dot{X}(t) = (\dot{x}(t) - y(t))\text{cos}(t) - (\dot{y}(t) + x(t) + \mu)\text{sin}(t) \tag{40}$$

$$\dot{Y}(t) = (\dot{x}(t) - y(t))\text{sin}(t) + (\dot{y}(t) + x(t) + \mu)\text{cos}(t) \tag{41}$$

B Analytic Gradients for Simple Shooting

The derivative of the objective function J with respect to the NLP variables: $\nabla J(\mathbf{x}) = \left[\dfrac{\partial J}{\partial \mathbf{x_0}}, \dfrac{\partial J}{\partial t_i}, \dfrac{\partial J}{\partial t_f}\right]^T$, and of the constraints, $\nabla \mathbf{c}(\mathbf{x})$, are implemented to improve the computational efficiency and accuracy of the optimization processes.

The spatial derivatives of the objective function $J(\mathbf{x})$ are expressed as:

$$\frac{\partial J}{\partial \mathbf{x_0}} = \frac{\partial J}{\partial \mathbf{x}_i} + \frac{\partial J}{\partial \mathbf{x}_f}\boldsymbol{\Phi}(t_0, t_f)$$

with:

$$\frac{\partial J}{\partial \mathbf{x}_i} = \frac{1}{\sqrt{(v_{x,i} - y_i)^2 + (v_{y,i} + x_i + \mu)^2}}\left[v_{y,i} + x_i + \mu;\ y_i - v_{x,i};\ v_{x,i} - y_i;\ v_{y,i} + x_i + \mu\right],$$

$$\frac{\partial J}{\partial \mathbf{x}_f} = \frac{1}{\sqrt{(v_{x,f} - y_f)^2 + (v_{y,f} + x_f + \mu - 1)^2}}\left[v_{y,f} + x_f + \mu - 1;\ y_f - v_{x,f};\ v_{x,f} - y_f;\ v_{y,f} + x_f + \mu - 1\right].$$

The time derivatives are calculated as

$$\frac{dJ}{dt_i} = -\frac{\partial \Delta V_2}{\partial \mathbf{x_f}}^T \boldsymbol{\Phi}(t_0, t_f) \cdot \mathbf{f}(\mathbf{x}_i, t_i) \quad ; \quad \frac{dJ}{dt_f} = \frac{\partial \Delta V_2}{\partial \mathbf{x_f}}^T \cdot \mathbf{f}(\mathbf{x}_f, t_f)$$

where

$$\frac{\partial \Delta V_2}{\partial \mathbf{x_f}} = \frac{1}{\sqrt{(v_{x,f} - y_f)^2 + (v_{y,f} + x_f + \mu - 1)^2}}\begin{bmatrix} v_{y,f} + x_f + \mu - 1 \\ y_f - v_{x,f} \\ v_{x,f} - y_f \\ v_{y,f} + x_f + \mu - 1 \end{bmatrix}$$

For the constraints:

$$\frac{\partial \boldsymbol{\psi}_\mathbf{i}}{\partial \mathbf{x}_i} = \begin{bmatrix} 2(x_i + \mu) & 2y_i & 0 & 0 \\ v_{x,i} & v_{y,i} & (x_i + \mu) & y_i \end{bmatrix}$$

$$\frac{\partial \boldsymbol{\psi}_\mathbf{f}}{\partial \mathbf{x}_f} = \begin{bmatrix} 2(x_f + \mu - 1) & 2y_f & 0 & 0 \\ v_{x,f} & v_{y,f} & (x_f + \mu - 1) & y_f \end{bmatrix}$$

From which:

$$\nabla \mathbf{c}(\mathbf{x}) = \begin{bmatrix} \dfrac{\partial \boldsymbol{\psi}_\mathbf{i}}{\partial x_i} & 0 & 0 \\ \dfrac{\partial \boldsymbol{\psi}_\mathbf{f}}{\partial x_f}\Phi_f & \dfrac{\partial \boldsymbol{\psi}_\mathbf{f}}{\partial x_f}(-\boldsymbol{\Phi}(t_0, t_f)\mathbf{f}(t_i)) & \dfrac{\partial \boldsymbol{\psi}_\mathbf{f}}{\partial x_f} \cdot \mathbf{f}(t_f)) \end{bmatrix}^T$$

where $\mathbf{f}$ is the right-hand side of the equations of motion and the STM $\mathbf{\Phi}$ is propagated with the variational approach:

$$\begin{cases} \dot{\mathbf{\Phi}}(t_0, t) = \mathbf{A}(t)\mathbf{\Phi}(t_0, t) \\ \mathbf{\Phi}0 = \mathbf{I}_{4\times 4} \end{cases}$$

The Jacobian matrix $\mathbf{A}$ of the equations of motion consists of partial derivatives of the equations of motion with respect to the state variables. The matrix A is defined as:

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial \dot{x}} & \frac{\partial f_1}{\partial \dot{y}} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial \dot{x}} & \frac{\partial f_2}{\partial \dot{y}} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial \dot{x}} & \frac{\partial f_3}{\partial \dot{y}} \\ \frac{\partial f_4}{\partial x} & \frac{\partial f_4}{\partial y} & \frac{\partial f_4}{\partial \dot{x}} & \frac{\partial f_4}{\partial \dot{y}} \end{bmatrix}$$

C Analytic Gradients for Multiple Shooting

This appendix provides the analytical expressions for the gradients of the objective function and the constraints used in the Multiple Shooting optimization method. This derviation follows the one presented in Appendix 2 (Topputo et al., 2019)[‡].

The derivative of the objective function J with respect to the optimization variables $\mathbf{x}$ is expressed as:

$$\frac{\partial J}{\partial \mathbf{x}} = \begin{bmatrix} P_1 & O & P_N & O \end{bmatrix},$$

where:

$$P_1 = \frac{1}{\sqrt{(v_{x,1} - y_1)^2 + (v_{y,1} + x_1 + \mu)^2}} \begin{bmatrix} v_{y,1} + x_1 + \mu \\ y_1 - v_{x,1} \\ v_{x,1} - y_1 \\ v_{y,1} + x_1 + \mu \end{bmatrix},$$

$$P_N = \frac{1}{\sqrt{(v_{x,N} - y_N)^2 + (v_{y,N} + x_N + \mu - 1)^2}} \begin{bmatrix} v_{y,N} + x_N + \mu - 1 \\ y_N - v_{x,N} \\ v_{x,N} - y_N \\ v_{y,N} + x_N + \mu - 1 \end{bmatrix}$$

The derivative of the equality constraints is expressed as:

$$\nabla \mathbf{c}_{eq}(\mathbf{x}) = \begin{bmatrix} \mathbf{\Phi}(t_1, t_2) & -\mathbf{I}_4 & O & O & Q_1^1 & Q_1^N \\ O & \mathbf{\Phi}(t_2, t_3) & -\mathbf{I}_4 & O & Q_2^1 & Q_2^N \\ O & O & \mathbf{\Phi}(t_3, t_4) & -\mathbf{I}_4 & Q_3^1 & Q_3^N \\ R_1 & O & O & O & O & O \\ O & O & O & R_N & O & O \end{bmatrix},$$

where:

$$Q_j^1 = -\frac{N-j}{N-1}\mathbf{\Phi}(t_j, t_{j+1})f(\mathbf{x}j, t_j) + \frac{N-j-1}{N-1}f(\varphi(\mathbf{x}_j, t_j, tj+1), t_{j+1}),$$

$$Q_j^N = -\frac{j-1}{N-1}\mathbf{\Phi}(t_j, t_{j+1})f(\mathbf{x}j, t_j) + \frac{j}{N-1}f(\varphi(\mathbf{x}_j, t_j, tj+1), t_{j+1}),$$

$$R_1 = \frac{\partial \psi_1}{\partial \mathbf{x}_1} = \begin{bmatrix} 2(x_1 + \mu) & 2y_1 & 0 & 0 \\ v_{x,1} & v_{y,1} & x_1 + \mu & y_1 \end{bmatrix},$$

[‡] K. Oshima, F. Topputo, T. Yanao, “Low-energy transfers to the Moon with long transfer time”, Celestial Mechanics and Dynamical Astronomy, Vol. 131, Article 4, 2019, DOI: 10.1007/s10569-019-9883-7.

$$R_N = \frac{\partial \boldsymbol{\psi}_N}{\partial \mathbf{x}_N} = \begin{bmatrix} 2(x_N + \mu - 1) & 2y_N & 0 & 0 \\ v_{x,N} & v_{y,N} & x_N + \mu - 1 & y_N \end{bmatrix}.$$

While the derivative of the inequality constraints can be expressed as:

$$\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} S_1 & O & O & O & O \\ O & S_2 & O & O & O \\ O & O & S_3 & O & O \\ O & O & O & \ddots & O \\ O & O & O & O & S_t \end{bmatrix},$$

where:

$$S_j := \frac{\partial \eta_j}{\partial x_j} = \begin{bmatrix} -2(x_j + \mu) & -2y_j & 0 & 0 \\ -2(x_j + \mu - 1) & -2y_j & 0 & 0 \end{bmatrix}, \quad j = 1, \ldots, N;$$

$$S_t := \begin{bmatrix} \frac{\partial \tau}{\partial t_1} & \ldots & \frac{\partial \tau}{\partial t_N} \end{bmatrix} = \begin{bmatrix} 1 & -1 \end{bmatrix}.$$

D Adimensionalization of Parameters

The parameters used in the trajectory optimization were adimensionalized to simplify the calculations and ensure consistency across units. The adimensionalization process involves scaling the quantities using the chosen distance, time, and mass units (`DU`, `TU`, `MU`). Below is the detailed procedure:

The fundamental units and constants are defined as:

```
% DATA:
data.TU = TU; % Time Unit
data.MU = MU; % Mass Unit
data.DU = DU; % Distance Unit
data.k1 = k1; % Debris spatial density constant 1
data.k2 = k2; % Debris spatial density constant 2
```

Using the above units, the physical parameters are converted into their adimensionalized forms:

```
% RESULTS:
data.Tmax = Tmax * ((TU^2)/(MU*DU)); % Maximum thrust
data.Isp = Isp / TU;                 % Specific impulse
data.m0 = m0 / MU;                   % Initial mass
data.g0 = g0 * (TU^2 / DU);          % Gravitational acceleration
data.mu = mu * (TU^2 / DU^3);        % Gravitational parameter
data.t0 = 0;                         % Initial time
data.rho0 = rho0 / DU;               % Reference radius
```

The initial and final states of the spacecraft are also adimensionalized as:

```
% Adimensionalized initial and target state:
x0 = [initialState; m0];
x0 = [x0(1:3)/DU; x0(4:6)*(TU/DU); x0(7)/MU]; % Initial state
xf = [finalState(1:3)/DU; finalState(4:6)*(TU/DU)]; % Final state
```

E Continuous Guidance Algorithm

Algorithm 1 Cost Function for Initial Costate and Final Time Estimation

```
Require: $[\boldsymbol{\lambda}0, tf]$, $[\mathbf{x}_{\text{target}}; 0; 0]$, $\mathbf{x}_0$
Ensure: res: Error vector.
  Initialize state vector and costate $[\mathbf{x}_0; \boldsymbol{\lambda}_0]$
  Initialize state transition matrix $\boldsymbol{\Phi}0$ as an identity matrix $\mathbf{I}_{14\times 14}$
  Construct complete initial state by appending $\boldsymbol{\Phi}_0$ to $[\mathbf{x}_0; \boldsymbol{\lambda}_0]$
  Propagate System and State Transition Matrix:
    Propagate the state from $t_0$ to $t_f$ by integrating the Euler-Lagrange equations and STM
    Extract final state vector $[\mathbf{x}_f; \boldsymbol{\lambda}_f]$ from the propagation results
    Extract final state transition matrix $\boldsymbol{\Phi}_f$ from the propagation results
  Compute Hamiltonian:
    Evaluate the Hamiltonian $H$ at the final state $[\mathbf{x}_f; \boldsymbol{\lambda}_f]$
  Compute the Derivatives:
    Use the final state transition matrix $\boldsymbol{\Phi}_f$ and final state to compute the gradient $\boldsymbol{\nabla}\mathbf{F}$
  Compute Error Vector:
  $\mathbf{res} := \begin{bmatrix} \mathbf{r}(t_f) - \mathbf{r}_f \\ \mathbf{v}(t_f) - \mathbf{v}_f \\ \lambda_m(t_f) \\ H(t_f) \end{bmatrix}$
  return res, $\boldsymbol{\nabla}\mathbf{F}$
```