SGN – Assignment #1

Alessandro Massini, 243670

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

This section is dedicated to find the coordinates of the five Lagrangian points in the contest of the 3D Earth-Moon Circular Restricted Three-Body Problem (CRTBP).

The coordinates have been computed starting from the scalar potential $\Omega(x, y, z)$ provided in Section 1 and by analyzing the equilibrium condition for the Libration Points:

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

From the theoretical framework, the Collinear Points (L1, L2, L3) can be determined by setting $y = z = 0$ and focusing solely on the partial derivative of the potential with respect to the x-component. This approach simplify the problem to a one dimensional analysis along the x-axis. Conversely, to identify the Triangular Points (L4, L5), the system of equations obtained by setting to zero the first and the second partial derivatives of Eq. (1) must be solved.

The numerical computation was implemented in MATLAB using the `fsolve` function as solver. To guarantee the required numerical precision, the `OptimalityTolerance` option was set to 10^{-11}. The initial guesses for the solver were carefully selected a priori based on the relative positions of the Libration points, ensuring that the algorithm consistently converged to a different point for each call.
The five Lagrangian Points resulted from the numerical analysis are shown in Fig. 1, alongside the unitary circles centered on the primaries and the partial derivative of the potential with respect to x-component, in the Earth-Moon rotating frame.

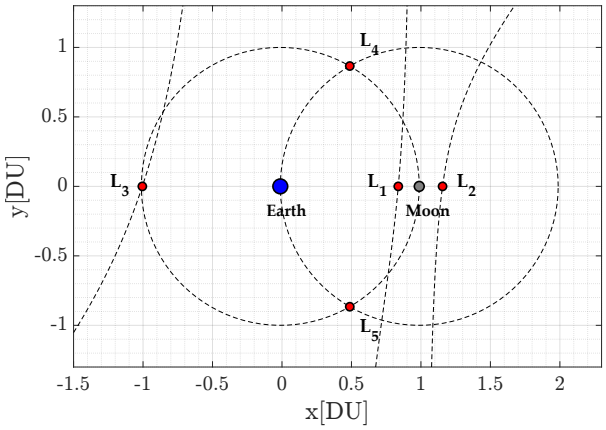

Figure 1: Lagrangian Points (@EMB Earth-Moon rotating frame).

In accordance with theoretical predictions, the Collinear Points are located along the Earth-Moon axis at the positions where the derivative of the potential with respect to the x-component equals zero, while the Triangular ones lie at the two intersections of unitary circles centered

The Jacobi Constant associated with each Lagrangian Point is then computed by exploiting their properties to remain fixed in the Earth-Moon rotating frame as they represents equilibrium points. Consequently, the expression for the Jacobi Constant simplifies to

$$C = 2\Omega(x, y, z) \tag{2}$$

The coordinates of each L-Point, along with the associated (C) are summarized in Table 1.

	L_1	L_2	L_3	L_4	L_5
x	+0.8369180073	+1.1556799131	−1.0050624018	+0.4878500000	+0.4878500000
y	0.0000000000	0.0000000000	0.0000000000	+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 purpose of the following section is to implement a differential correction scheme to refine the provided initial guess and determine periodic Halo Orbit characterized by $C = 3.09$.

The first step involved verifying the periodicity of the provided initial state. This was achieved propagating the state using a numerical CRTBP integrator with a sufficiently large integration time. The resulting trajectory was analyzed graphically to determine whether, after one revolution, the orbit returned to its initial condition. Fig. 2 exhibits the divergence of the orbit, indicting that the initial state does not correspond to a periodic Halo Orbit.

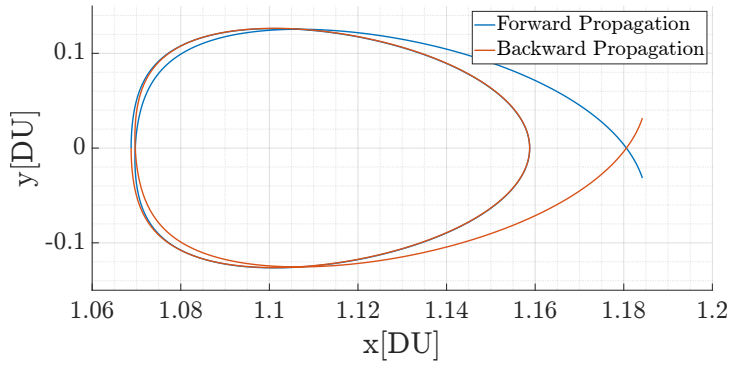

Figure 2: Forward integration of the initial state to $t_f = 5[DU]$ (@EMB EMRF).

A theoretical framework was put into practice in order to achieve the desired periodic orbit. To find the perpendicular crossing of the $y = 0$ plane, an event function was first added to the propagator. This function was used to identify the time t_f at which the trajectory crosses the plane, providing the initial guess for the following analysis. Subsequently, a differential correction scheme was developed exploiting the first-order expansion of the state's flow with respect to the initial state and the final integration time. This procedure iteratively adjusts the initial state and the final integration time based on the errors observed in the final state, ensuring convergence toward a periodic solution. Lastly, an enhanced linear system that included an additional constraint in the correction procedure was put into place to guarantee that the condition on the Jacobi Constant was likewise fulfilled.

As first step, in Eq. (3) are presented the Taylor expansion of the flow and of the Jacobi Constant:

$$\begin{aligned} \boldsymbol{\varphi}(\mathbf{x_0} + \boldsymbol{\delta}\mathbf{x_0}, t_f + \delta t_f) &= \boldsymbol{\varphi}(\mathbf{x_0}, t_f) + \frac{\partial \boldsymbol{\varphi}(\mathbf{x_0}, t_f)}{\partial \mathbf{x_0}} \boldsymbol{\delta}\mathbf{x_0} + \frac{\partial \boldsymbol{\varphi}(\mathbf{x_0}, t_f)}{\partial t_f} \delta t_f \\ J(\mathbf{x_0} + \boldsymbol{\delta}\mathbf{x_0}) &= J(\mathbf{x_0}) + \frac{\partial J(\mathbf{x_0})}{\partial \mathbf{x_0}} \boldsymbol{\delta}\mathbf{x_0} \end{aligned} \tag{3}$$

where $\dfrac{\partial \boldsymbol{\varphi}(\mathbf{x_0}, t_f)}{\partial t_f}$ is the derivative of the state at the final time $\dot{\mathbf{x}}(t_f)$ and $\dfrac{\partial \boldsymbol{\varphi}(\mathbf{x_0}, t_f)}{\partial \mathbf{x_0}}$ is the state transition matrix evaluated at the final time $\boldsymbol{\Phi}(t_0, t_f)$. The state transition matrix was computed through a variational approach in order to ensure numerical accuracy in the computation:

$$\begin{cases} \dot{\mathbf{x}}(\mathbf{t}) = \mathbf{f}(\mathbf{x}, t) & \text{Initial Condition: } \mathbf{x}(t_0) = \mathbf{x_0} \\ \dot{\boldsymbol{\Phi}}(\mathbf{x_0}, t_0, t) = \dfrac{\partial \mathbf{f}(\mathbf{t})}{\partial \mathbf{x}} \boldsymbol{\Phi}(t_0, t) & \text{Initial Condition: } \boldsymbol{\Phi}(t_0, t_0) = \mathbf{I}_{6\times 6} \end{cases} \tag{4}$$

The differential correction system is established by recognizing that the left-hand side (LHS) of Eq. (3) represents the desired final state (and C) after propagation, while the first terms on the right-hand sides (RHS) of the two equations correspond to the actual final state (and C) obtained from the propagation starting from the non-corrected initial state.

The desired state values are those that ensure a perpendicular crossing of the $y = 0$ plane. Specifically, this requires $y = 0$, $v_x = 0$, and $v_z = 0$. Additionally, the Jacobi Constant is enforced to have the value $C = 3.09$, ensuring consistency with the requirements.
The needed corrections are then determined by solving the 7×7 linear system in Eq. (5) where $\mathbf{e}$ represents the vector of errors between the desired and the actual final values, and $\boldsymbol{\delta}$ denotes the vector of corrections to be applied to the initial state of the system.

$$\mathbf{e} = \begin{bmatrix} \boldsymbol{\Phi}(t_0, t_f) & \dot{\mathbf{x}}(t_f) \\ \boldsymbol{\nabla} J_x^{\top}(\mathbf{x_0}) & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}\mathbf{x_0} \\ \delta t_f \end{bmatrix} \tag{5}$$

Since the 7×7 system, characterized by the four constraints on y_f, v_{x_f}, v_{z_f}, and C, and the four free variables x_0, z_0, v_{y_0}, and t_f, was inherently under-determined, its reduction to a 4×4 system became necessary to guarantee a unique solution. This latter was performed by selecting the rows of the system matrix associated with the constraint components and the columns corresponding to the adjustable initial variables.

Once the system was solved and the appropriate corrections were found, the final step of the algorithm focused on refining the initial state through an iterative update process. The formula that encapsulates all these steps and serves as the core of the algorithm is:

$$\begin{bmatrix} x_0 \\ z_0 \\ v_{y_0} \\ t_f \end{bmatrix}_{k+1} = \begin{bmatrix} x_0 \\ z_0 \\ v_{y_0} \\ t_f \end{bmatrix}_{k} - \begin{bmatrix} \boldsymbol{\Phi}_{21} & \boldsymbol{\Phi}_{23} & \boldsymbol{\Phi}_{25} & \dot{y}_f \\ \boldsymbol{\Phi}_{41} & \boldsymbol{\Phi}_{43} & \boldsymbol{\Phi}_{45} & \dot{v}_{x_f} \\ \boldsymbol{\Phi}_{61} & \boldsymbol{\Phi}_{63} & \boldsymbol{\Phi}_{65} & \dot{v}_{z_f} \\ \frac{\partial J}{\partial x} & \frac{\partial J}{\partial z} & \frac{\partial J}{\partial v_y} & 0 \end{bmatrix}^{-1} \begin{bmatrix} y_f - 0 \\ v_{x_f} - 0 \\ v_{z_f} - 0 \\ C_f - 3.09 \end{bmatrix} \tag{6}$$

The reduced linear system was solved using MATLAB `mldivide` function and the initial state was iteratively updated until all the componennts of the error vector fell below a predefined tolerance of 10^{-12}.

The corrected initial state is summarized in Table 2, while in Fig. 3 is shown the obtained periodic Halo Orbit in Earth-Moon rotating reference frame.

x	y	z	v_x	v_x	v_x
1.0590402077	0.0000000000	0.0739277378	0.0000000000	0.3469245709	0.0000000000

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

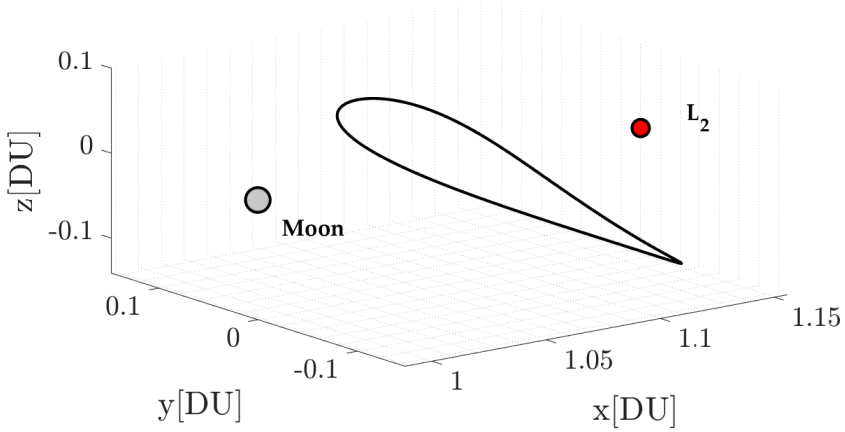

Figure 3: Periodic Halo Orbit with $C = 3.09$ (@EMB EMRF).

1.3 Numerical Continuation

As the final step of the analysis, a procedure was developed to determine the correct state for a periodic Halo Orbit with $C = 3.04$. Given that a solution was previously obtained for $C = 3.09$ (as shown in Section 1.2), the numerical continuation technique was exploited to enhance the convergence of the algorithm and adapt the solution to the new Jacobi Constant.

A family of $n = 21$ Halo Orbit, with Jacobi Constants equally spaced between $C = 3.09$ and $C = 3.04$, was generated and solved iteratively. At each iteration, the algorithm took in input the initial guess of the previous step and updated it applying it the corrections computed through the procedure outlined in Section 1.2. This iterative process continued until the final Halo Orbit, corresponding to $C = 3.04$, was found. The corrected initial state is showcased in Table 3.

x	y	z
1.0125655235	0.0000000000	0.0672339583
v_x	v_y	v_x
0.0000000000	0.5103251959	0.0000000000

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

In Fig. 4 the family of Halo Orbit generated with Jacobi Constant values ranging from $C = 3.09$ to $C = 3.04$ is depicted, alongside the moon and the L2 point.

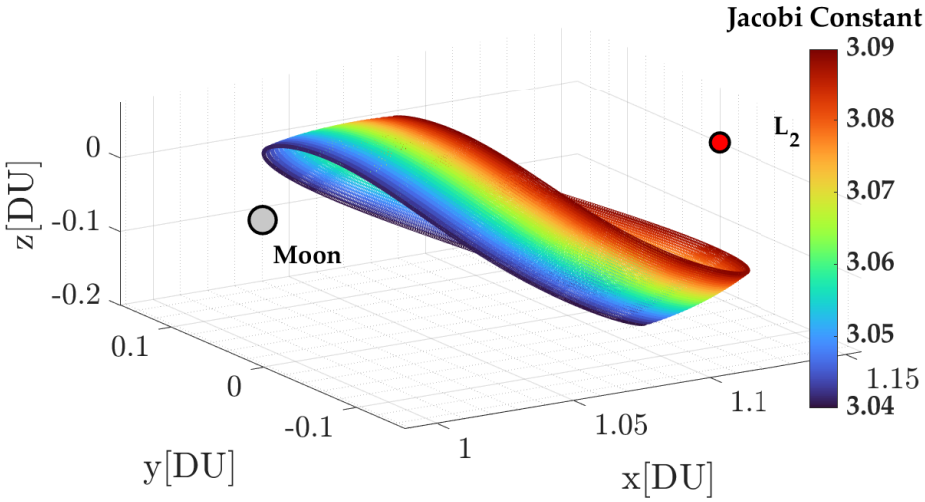

Figure 4: Families of periodic Halo Orbits from $C = 3.09$ to $C = 3.04$(@EMB EMRF).

As shown in Fig. 4, orbits with lower values of the Jacobi Constant are larger in size. This occurs because a decrease in C corresponds to an increase in the mechanical energy associated with the orbit:

$$C = -\frac{E}{2} \tag{7}$$

allowing the trajectory to extend its allowable region and moving further from the corresponding Lagrangian point.

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 Initial Guess Generation

The purpose of the exercise is to optimize a two impulses Earth-Moon transfer in the restricted four-body problem with the Earth, the Moon and the Sun as primaries.
The analysis began with the generation of a first guess solution exploiting $(\alpha, \beta, \delta, t_i)$ provided in Section 2 and the methodology in (Topputo, 2013)[2]. The characteristic mass ratio (μ) of the problem was retrieved from the gravitational constants of the Earth and the Moon using `cspice_bodvrd`. Subsequently, the adimensionalized radius of the parking orbit (r_i) was computed from Earth's radius (R_E), the provided altitude (h_i) and the distance unit (DU). Being $v_0 = \beta\sqrt{\dfrac{1-\mu}{r_0}}$ the initial guess $\mathbf{x_0}$ is computed as:

$$\begin{aligned} x_0 &= r_0 \cos\alpha - \mu & y_0 &= r_0 \sin\alpha \\ \dot{x}_0 &= -(v_0 - r_0)\sin\alpha & \dot{y}_0 &= (v_0 - r_0)\cos\alpha \end{aligned} \tag{8}$$

The initial adimensionalized state is summarized 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 was then propagated over the time interval t_i to $t_i + \delta$ using a PBRFBP propagator and rotated in Earth-Centered Inertial (ECI) reference frame following the procedure outlined in [2]. In Fig. 5 and Fig. 6 is depicted the resulting trajectory, displayed both in the Earth-Moon rotating frame (EMB) and in ECI one.

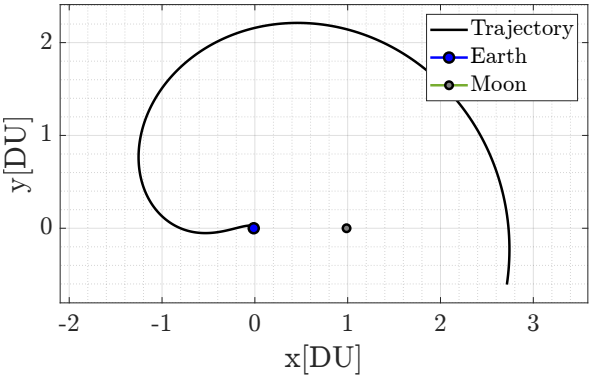

Figure 5: Initial Trajectory (@EMB EMRF).

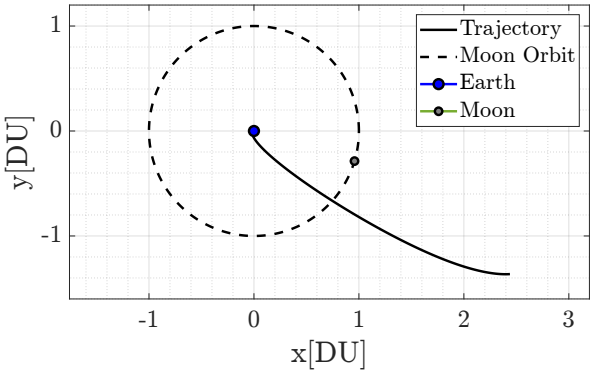

Figure 6: Initial Trajectory (@Earth ECI).

2.2 Simple Shooting

The first method employed to solve the two impulses Earth-Moon transfer optimization problem is the simple shooting technique. The optimization variable are set to be $\{\mathbf{x_i}, t_i, t_f\}$ and the initial guess for the optimizer is the one computed in Section 2.1. The statement of the problem is defined as follows.

Statement of the Problem [2]: Let $\mathbf{x}(t) = \boldsymbol{\varphi}(\mathbf{x}_i, t_i; t)$ be a solution of the PBRFBP equations of motion, integrated from $(\mathbf{x}_i, t_i)$ to $t \geq t_i$. The optimization problem for two-impulse Earth–Moon transfers consists of finding $\{\mathbf{x}_i, t_i, t_f\}$, $t_f > t_i$, such that $\boldsymbol{\psi}_i(\mathbf{x}_i) = 0, \boldsymbol{\psi}_f(\mathbf{x}_f) = 0$, where $\mathbf{x}_f = \boldsymbol{\varphi}(\mathbf{x}_i, t_i; t_f)$, and the function

$$\Delta v(\mathbf{x}_i, t_i, t_f) = \Delta v_i(\mathbf{x}_i) + \Delta v_f(\boldsymbol{\varphi}(\mathbf{x}_i, t_i; t_f)) \tag{9}$$

is minimized.

In which $\boldsymbol{\psi}_i(\mathbf{x}_i)$ and $\boldsymbol{\psi}_f(\mathbf{x}_f)$ are respectively the non-linear equality constraints to satisfy at the Earth and at the Moon and are defined as:

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

The costs for the initial and final maneuvers, $\Delta v_i(\mathbf{x}_i)$ and $\Delta v_f(\mathbf{x}_f)$ can be quantified by:

$$\begin{aligned}\Delta v_i &= \sqrt{(\dot{x}_i-y_i)^2+(\dot{y}_i+x_i+\mu)^2}-\sqrt{\frac{1-\mu}{r_i}}\\ \Delta v_f &= \sqrt{(\dot{x}_f-y_f)^2+(\dot{y}_f+x_f+\mu-1)^2}-\sqrt{\frac{\mu}{r_f}}\end{aligned} \tag{11}$$

The optimization problem is finally mapped into a Nonlinear Programming (NLP) problem, with variables $\{\mathbf{x}_i, t_i, t_f\}$ state as follows:

$$\min_{\mathbf{x}_i,t_i,t_f} \Delta v(\mathbf{x}_i,t_i,t_f) \quad \text{s.t.} \quad \mathbf{c}(\mathbf{x}_i,t_i,t_f)=\mathbf{0} \quad \text{where:} \quad \mathbf{c}=\begin{cases}\boldsymbol{\psi}_i(\mathbf{x}_i)\\ \boldsymbol{\psi}_f(\boldsymbol{\varphi}(\mathbf{x}_i,t_i;t_f))\end{cases} \tag{12}$$

The NLP problem was solved using MATLAB's `fmincon` function, with the `active-set` algorithm selected after several trials due to its superior performance. The `ConstraintTolerance` was set to 10^{-10} to ensure high accuracy in the solution.

In the first place, the optimization was carried out without providing analytical gradients for either the objective function or the constraints. As a result, the solver computed the required derivatives internally using the Central Finite Differences method.
The resulting trajectory is depicted in Fig. 7, in both the Earth-Moon Rotating Frame (EMB) and the Earth-Centered Inertial Frame (ECI).

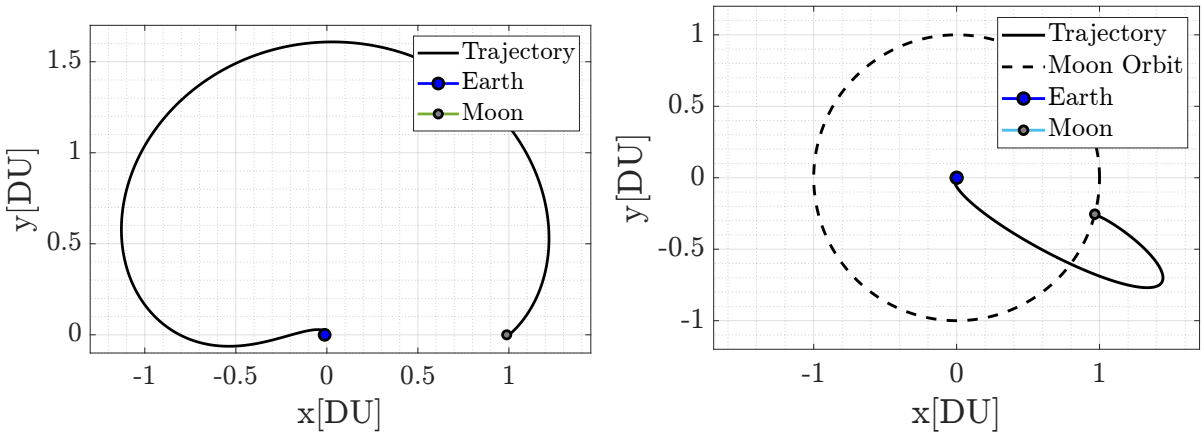

Figure 7: Simple Shooting trajectory without gradients (@EMB EMRF and @Earth ECI).

A second optimization was conducted by supplying the analytic gradients of both the objective function and the constraints with respect to the problem's variables. The State Transition Matrix (STM), required for the gradient computation, was derived using a variational approach, thereby ensuring a higher level of accuracy in the results. The full derivation of the methodology and the comprehensive set of formulas are provided in Appendix A.

The integration of the analytic gradients improved significantly the performance of `fmincon`, leading to a faster convergence both in terms of time and iterations which decreased from 21 to 14, while function evaluations dropped from 190 to just 31. Furthermore, as highlighted by

Table 6, it resulted in a lower cost function, that the optimizer not only converged faster but also achieved a better solution compared to the previous case.
The resulting trajectory from the second analysis is showcased in Fig. 9, visualized in both the Earth-Moon Rotating Frame (EMB) and the Earth-Centered Inertial Frame (ECI).

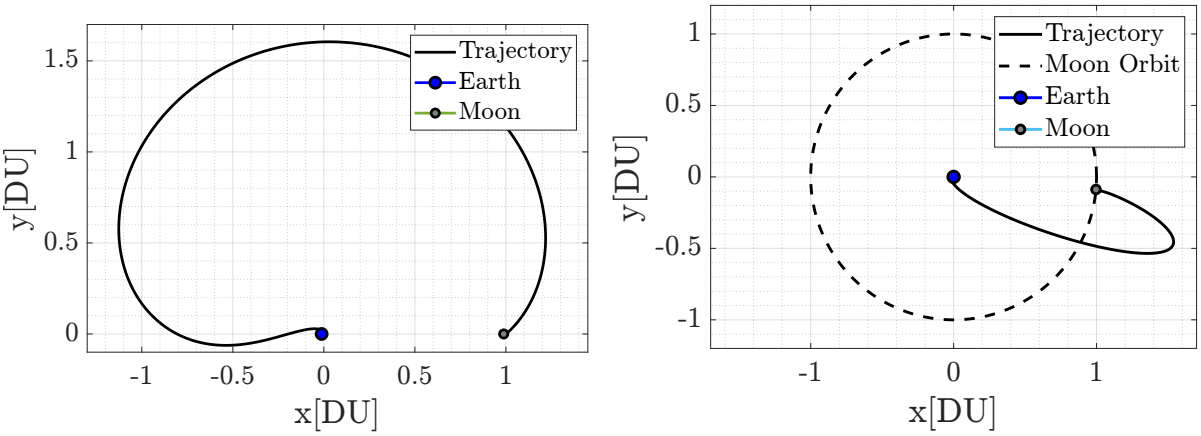

Figure 8: Simple Shooting trajectory with gradients (@EMB EMRF and @Earth ECI).

The optimal solutions for both approaches, with and without analytic gradients, are presented in Table 6, along with the respective cost function values.

Gradients	$r_{x,0}$ $[DU]$	$r_{y,0}$ $[DU]$	$v_{x,0}$ $[VU]$	$v_{y,0}$ $[VU]$	t_i $[TU]$	t_f $[TU]$	$\Delta v[VU]$
False	0.001610	0.010027	−6.299679	8.645478	2.026	6.033	4.00755
True	0.001620	0.010013	−6.290741	8.651415	2.198	6.203	4.00683

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

The results of the two methods were validated by computing the dimensionalized position and velocity errors at the Moon's arrival. As shown in Table 7, both methods yield very low errors, indicating a high level of optimization accuracy. However, such high precision should be interpreted with caution as the PBRFBP dynamics model, while capable to catch the key insights of the problem, relies on several assumptions, such as planar motion and perfectly circular orbits for the primary bodies, neglecting the effects of orbital ellipticity and inclination. Furthermore, the model does not take into account the uncertainties in the initial state estimation and their propagation along the trajectory. All these factors, inherent in a real-world scenario, would impact significantly on the accuracy of the trajectory, resulting in larger deviations than those observed in the simulation.

Gradients	$\lvert\boldsymbol{\psi}_r(\mathbf{x_f})\rvert$ $[m]$	$\lvert\boldsymbol{\psi}_v(\mathbf{x_f})\rvert$ $[m/s]$
False	$4.612114e-06$	$6.347395e-09$
True	$9.605503e-06$	$1.439946e-08$

Table 7: Dimensionalized Errors at the Moon Parking Orbit.

2.3 Multiple Shooting

In order to further enhance the accuracy and the robustness of the solution the optimization problem was also approached with a Multiple Shooting technique.

For this purpose, the time interval was divided into $N = 4$ equally spaced nodes. The initial guess derived in Section 2.1 was discretized over the time grid such that $\mathbf{x}_j = \mathbf{x}(t_j)$,

where $j = 1, \ldots, N$. These resulting states, $\mathbf{x}_j$, were then treated as the initial variables of the problem, forming the following set:

$$\mathbf{y} = \{\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, \mathbf{x_4}, t_1, t_4\} \tag{13}$$

Differently from Simple Shooting, where the propagation is performed directly from t_i to t_f, Multiple Shooting divides the trajectory into $N-1$ segments and performs propagations over the individual time intervals $[t_j, t_{j+1}]$, using $\mathbf{x}_j$ as the initial condition for each segment. Continuity in position and velocity at the boundaries of these intervals is ensured by enforcing a set of equality constraints $\boldsymbol{\zeta_i}$, known as defects and defined as:

$$\boldsymbol{\zeta}_j = \boldsymbol{\varphi}(\mathbf{x}_j, t_j; t_{j+1}) - \mathbf{x}_{j+1}, \quad j = 1, ..., N-1 \tag{14}$$

By adding the boundary conditions (Eq. (10)) for Earth departure and Moon arrival, the augmented set of equality constraints for the Multiple Shooting method is expressed as:

$$\mathbf{c}(\mathbf{y}) = \{\boldsymbol{\zeta_1}, \ldots, \boldsymbol{\zeta_{N-1}}, \boldsymbol{\psi_1}, \boldsymbol{\psi_N}\}. \tag{15}$$

Furthermore, in order to avoid impacts with Earth and Moon, inequality constraints are defined at each node

$$\boldsymbol{\eta}_j := \begin{bmatrix} R_e^2 - (x_j + \mu)^2 - y_j^2 \\ R_m^2 - (x_j + \mu - 1)^2 - y_j^2 \end{bmatrix} < \mathbf{0}, \quad j = 1, \ldots, N, \tag{16}$$

where R_e and R_m are the non dimensional Radii of Earth and Moon. To ensure consistency, an equality condition on inital and final time $\tau := t_i - t_f < 0$ is also set goinf to define the total inequality constraints of the problem as:

$$\mathbf{g}(\mathbf{y}) = \{\boldsymbol{\eta}_j, ..., \boldsymbol{\eta}_N, \tau\} \tag{17}$$

Finally, the objective function of the problem is defined is defined in Eq. (9), with the only difference tt no longer depends directly on time, as $\mathbf{x}_f = \mathbf{x}_N$ is is now treated as a variable of the problem.

Having defined all the quantity of interest it straightforward to verify that $\mathbf{y}$ is $(4N + 2)$-dimensional, $\mathbf{c}$ is $4N$-dimensional and $\mathbf{g}$ is $(2N + 1)$-dimensional. Consequently, with the inequality constraints satisfied, the problem is well-posed. The final formulation of the NLP problem for the Multiple Shooting is as follows:

$$\min_{\mathbf{y}} f(\mathbf{y}) \quad \text{s.t.} \quad \begin{cases} \mathbf{c}(\mathbf{y}) = 0 \\ \mathbf{g}(\mathbf{y}) < 0 \end{cases} \tag{18}$$

The NLP problem was solved using MATLAB's `fmincon` function, with the `active-set` algorithm. The `ConstraintTolerance` was increased to 10^{-7} to ensure a good trade-off between accuracy of the solution and computational efficiency in attaining convergence. Moreover, analytic gradients for the cost function, as well as the equality and inequality constraints, were supplied to the solver in order to further enhance its convergence performances. The complete derivation of the gradients is detailed in Appendix B.
In Table 8 are reported the dimensional errors at the nodes of the optimization.

Error	At x_i	At x_f	At x_2	At x_3	At x_4
Pos $[m]$	$9.1181e-05$	$4.5462e-08$	26.7338	0.2219	0.1667
Vel $[m/s]$	$-1.0414e-10$	$-6.5095e-12$	$1.0599e-04$	$1.1194e-06$	$7.5934e-05$

Table 8: Multiple Shooting dimensional errors at the nodes.

As shown in the table, despite the increase in `ConstraintTolerance`, the errors at departure and arrival remain largely consistent with those of the Simple Shooting case, yielding results that are highly satisfactory. However, the norms of the defects at the nodes are slightly larger, in the order of meters and meters per seconds. While these could be reduced by lowering the `ConstraintTolerance`, doing so would, as previously noted, significantly increase the computational time required for convergence. Overall, the results achieved provide an excellent compromise between computational efficiency and accuracy.

The optimal solution of the Multiple Shooting optimization is summarized in Table 9 in terms of initial state and tranfer time, along with the total Δv of the transfer.

$r_{x,0}$ $[DU]$	$r_{y,0}$ $[DU]$	$v_{x,0}$ $[VU]$	$v_{y,0}$ $[VU]$	t_i $[TU]$	t_f $[TU]$	$\Delta v[VU]$
0.004074	0.005165	−3.244084	10.190529	2.4644	6.1289	3.99731

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

The total cost of the transfer is further reduced with respect to the Simple Shooting optimization confirming the superior accuracy of the Multiple Shooting method, especially when addressing nonlinearities and extended transfer durations.

The resulting optimized trajectory from the Multiple Shooting analysis is showcased in Fig. 9, visualized in both the Earth-Moon Rotating Frame (EMB) and the Earth-Centered Inertial Frame (ECI).

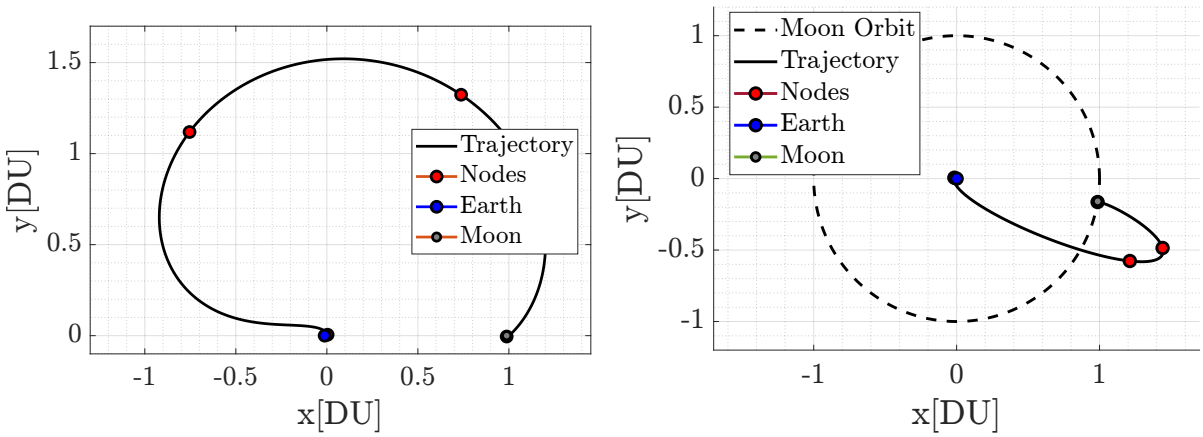

Figure 9: Multiple Shooting @EMB EMRF (left) and @Earth ECI (right)

2.4 N-body Propagator

Since the optimization relied on a propagator based on the Planar Bicircular Restricted Four-Body Problem, which includes certain approximations such as planar motion and circular orbits of the primaries, the solution was validated by propagating the optimal state (with gradients) found in Section 2.2 using an N-body propagator. This approach provides a more realistic representation of real-world scenarios, accounting for additional gravitational influences and orbital dynamics.

The N-body propagator relies on the ephemeris time to determine the relative positions of celestial bodies at a given instant. Therefore, to initialize the N-body propagation, it was necessary to identify the initial epoch where the relative positions of the Sun and Moon align with those assumed during the optimization process. A zero finding problem was then formulated to align the Moon-EMB-Sun angle to the initial angle $\vartheta_0 = \omega_s t_i$ where ω_s represents the scaled angular velocity of the sun relative to the EMB. As stated in Section 2, this angle is

approximately periodic with the Moon's revolution period. Therefore, the search for the initial epoch was restricted to this time interval starting from `2024 Sep 28 00:00:00.000 TDB`.

The problem was solved using MATLAB's `fzero` function. Once the initial epoch was identified, the time of flight obtained from Table 6 (with gradients) was added to determine the final epoch. The initial and final epoch retrieved are summarized in Table 10 along with the dimensional initial state in ECI obtained through the same procedure outlined in [2].

Symbol	**Calendar epoch** (UTC)
t_i	2024-10-12T11:40:25.186
t_f	2024-10-29T21:05:11.333

$r_{x,0}$ $[km]$	$r_{y,0}$ $[km]$	$r_{z,0}$ $[km]$
-6223.6298	2026.1404	0.0

$v_{x,0}$ $[km/s]$	$v_{y,0}$ $[km/s]$	$v_{z,0}$ $[km/s]$
-3.39799	-10.43750	0.0

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

In order to perform the propagation, the N-body dynamics has to be defined. Let $\mathbf{x} = [\mathbf{r}, \mathbf{v}]$ be the left-hand side of the equations of motion and $\mathbf{a}_0 = -\dfrac{\mu_E \mathbf{r}}{r^3}$ the gravitational acceleration of the central body (Earth in this case) the N-body dynamics is defined as follows:

$$\mathbf{x} = \begin{bmatrix} \mathbf{v} \\ \mathbf{a}_0 + \sum_{i \neq 0} \mathbf{a}_i \end{bmatrix} \quad \text{with} \quad \mathbf{a}_i = -\frac{\mu_i}{|\mathbf{r} - \boldsymbol{\rho}_i|^3} \left(\mathbf{r} + \boldsymbol{\rho}_i f(q) \right) \tag{19}$$

in which $\mathbf{a}_i$ represent the gravitational contribution of the i-th perturbing body, $\boldsymbol{\rho}_i$ is its position vector with respect to the center and $f(q)$ is defined as:

$$f(q) = \frac{q(3 + 3q + q^2)}{1 + (1 + q)^{3/2}} \quad \text{with} \quad q = \frac{\mathbf{r} \cdot (\mathbf{r} - 2\boldsymbol{\rho}_i)}{|\boldsymbol{\rho}_i|^2} \tag{20}$$

The obtained trajectory after the N-body propagation has been compared to the one derived from the Simple Shooting method with gradients. The results are shown in Fig. 10.

It is clear that while the two trajectories are similar, they are not identical. The first noticeable difference is that the PBRFBP trajectory lies entirely within the Moon's orbital plane, whereas the N-body trajectory, being computed in a three-dimensional framework, exhibits out-of-plane position and velocity different from zero. Secondly, the PBRFBP trajectory intersects the Moon precisely, while the N-body trajectory falls short of reaching it. This discrepancy likely arises because the initial state used for the N-body propagation was optimized using a different dynamical model, making slight variations in the trajectory an expected outcome.

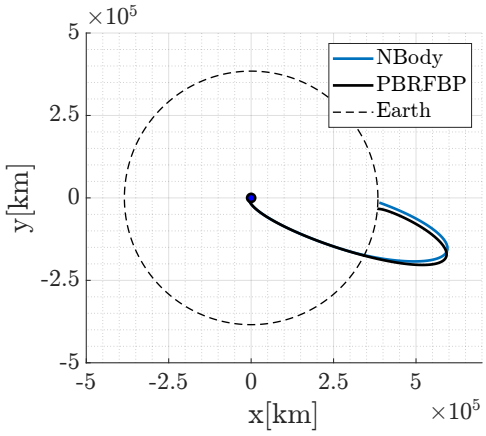

Figure 10: Comparison between N-body propagator and PBRFBP propagator @Earth ECI

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

3.1 Debris Spatial Density and Target state Computation

The design of an optimal orbit raising maneuver, that minimizes the risk of impact exploiting a low thrust option, begins with the analysis of the debris spatial density function provided in Section 3. The function variation with respect to the altitude of the orbit is depicted in Fig. 11, covering the range in the interval $[h_i - 100; h_f + 100]$.

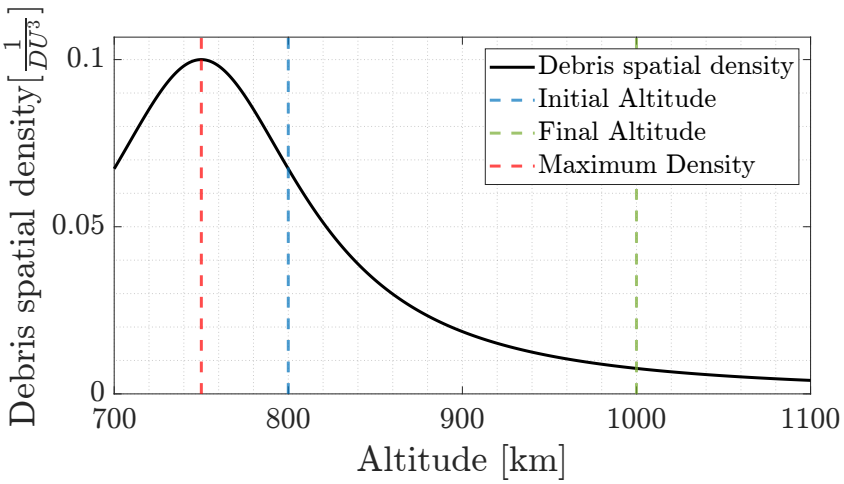

Figure 11: Debris spatial density as a function of the distance from Earth center.

From the figure, it can be observed that the maximum value of the debris spatial density function occurs below the initial orbital altitude. Consequently, the function exhibits a monotonically decreasing behavior over the interval $[r_i; r_f]$, indicating that the density of debris is diminishing throughout the transfer trajectory to be designed.

Subsequently, the analysis focuses on defining the initial and final states of the transfer. Leveraging the circular nature of the orbits and the fact that both states lie along the x-axis of the J2000 equatorial frame, it was determined that the x-coordinate is the only non-zero position component for both states. Additionally, the velocity components of the final state were derived based on the inclination change, which introduced a non-zero z-component of velocity, alongside the tangential y-component. The resulting states equations are reported in Eq. (21), while their dimensional values in Table 12.

$$\mathbf{x}_i = \begin{bmatrix} R_e + h_i \\ 0 \\ 0 \\ 0 \\ \sqrt{\dfrac{\mu_e}{R_e + h_i}} \\ 0 \end{bmatrix} \qquad \mathbf{x}_f = \begin{bmatrix} R_e + h_f \\ 0 \\ 0 \\ 0 \\ \sqrt{\dfrac{\mu_e}{R_e + h_f}} \cos(\Delta i) \\ \sqrt{\dfrac{\mu_e}{R_e + h_f}} \sin(\Delta i) \end{bmatrix} \tag{21}$$

$r_{x,i}$ $[km]$	$r_{y,i}$ $[km]$	$r_{z,i}$ $[km]$	$v_{x,i}$ $[km/s]$	$v_{y,i}$ $[km/s]$	$v_{z,i}$ $[km/s]$
7178.136600	0	0	0	7.45183148	0

$r_{x,f}$ $[km]$	$r_{y,f}$ $[km]$	$r_{z,f}$ $[km]$	$v_{x,f}$ $[km/s]$	$v_{y,f}$ $[km/s]$	$v_{z,f}$ $[km/s]$
7378.136600	0	0	0	7.34950906	0.09621034

Table 12: Initial and target state @Earth J2000 frame.

3.2 Adimensionalized Parameters

The adimensionalization of the parameters was carried out using the values of DU (Distance Unit) and MU (Mass Unit) provided in Table 11 and enforcing that $\mu_{scaled} = 1$. Knowing that μ has units of $\left[\frac{km^3}{s^2}\right]$, it was possible to retrieve the Time Unit (TU) and the Velocity Unit (VU) as follows:

$$TU = \sqrt{\frac{\mu_{scaled} DU^3}{\mu}}, \qquad VU = \frac{DU}{TU} \tag{22}$$

In Table 13, the adimensionalized parameters used to formulate the problem in the subsequent sections are presented, alongside the constants employed for the adimensionalization process.

$DU\ [km]$	$MU\ [kg]$	$TU\ [s]$	$VU\ \left[\frac{km}{s}\right]$
7178.1366	1000	963.2715	7.4518

$\mu\ \left[\frac{DU^3}{TU^2}\right]$	$\rho_0[DU]$	$m_0[MU]$	$T_{\max}\ \left[\frac{MU\,DU}{1000 \cdot TU^2}\right]$	$I_{sp}[TU]$	$g_0\ \left[\frac{DU}{1000 \cdot TU^2}\right]$
1	0.99303	1	3.8780e-04	3.23896	1.2681

Table 13: Adimensionalized Parameters

The adimensionalized initial and final states are shown in Eq. (23)

$$\mathbf{x}_i = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \qquad \mathbf{x}_f = \begin{bmatrix} 1.027862 \\ 0 \\ 0 \\ 0 \\ 0.986269 \\ 0.012911 \end{bmatrix} \tag{23}$$

3.3 PMP Formulation

The continuous guidance problem has been formulated in accordance with the Pontryagin Maximum Principle (PMP). A key advantage of this approach lies in the PMP's ability to inherently embeds restrictions on the control vector $\mathbf{u}$ directly within the problem formulation.

The objective function, as defined in Section 3, is structured to minimize the probability of collision by reducing the duration of the transfer within regions of high debris density. Accordingly, the problem is treated as a time-optimal control problem. The problem can be then formulated as follows.

$$\min_{(\hat{\boldsymbol{\alpha}},u)\in\Omega} \int_{t_0}^{t_f} l(\mathbf{x},\mathbf{u},t)\,dt \quad \text{s.t.} \quad \begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{u},t), \\ \mathbf{x}(t_0) = \mathbf{x}_0, \\ \mathbf{r}(t_f) = \mathbf{r}_f, \\ \mathbf{v}(t_f) = \mathbf{v}_f, \\ \lambda_m(t_f) = 0. \end{cases} \tag{24}$$

where:

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

and Ω is the set of admissible control actions, $\hat{\alpha}$ is the thrust pointing unit vector and u is the thrust throttle factor:

$$\Omega = \{(u,\hat{\boldsymbol{\alpha}}) : u \in [0,1], ||\hat{\boldsymbol{\alpha}}|| = 1\} \tag{26}$$

The dynamics of the state $\mathbf{f}(\mathbf{x}, \mathbf{u}, t)$ is defined, in the context of the two-body problem model as:

$$\mathbf{f}(\mathbf{x}, \mathbf{u}, t) = \begin{cases} \dot{\mathbf{r}} = \mathbf{v}, \\ \dot{\mathbf{v}} = -\dfrac{\mu}{r^3}\mathbf{r} + u\dfrac{T_{\max}}{m}\hat{\boldsymbol{\alpha}}, \\ \dot{m} = -u\dfrac{T_{\max}}{I_{\text{sp}} g_0}. \end{cases} \tag{27}$$

In order to define in a compact way the necessary conditions for the optimality, it is convenient to define the Hamiltonian. Let $\boldsymbol{\lambda}$ be a n-dimensional (in this case $n = 7$) vector of costate multipliers of the dynamics, the Hamiltonian is:

$$H = l + \boldsymbol{\lambda}^{\top}\mathbf{f} \tag{28}$$

From Calculus Variation theory and the application of the PMP, the necessary conditions for the optimality are the so called Euler-Lagrange equations:

$$\begin{cases} \dot{\mathbf{x}} = \dfrac{\partial H}{\partial \boldsymbol{\lambda}} \\ \dot{\boldsymbol{\lambda}} = -\dfrac{\partial H}{\partial \mathbf{x}} \\ (u^*, \hat{\boldsymbol{\alpha}}^*) = \arg\min\limits_{\mathbf{u}\in\Omega} H(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{u}, y) \end{cases} \tag{29}$$

in this way, the PMP establishes that the control variable must be selected to optimize the Hamiltonian at each time instant within its admissible domain Ω.

With the necessary conditions and the maximum principle defined, the analysis can proceed to address the given optimal orbit raising. As previously mentioned, the transfer can be approached as a time-optimal control problem. By applying the PMP, the following formulation is obtained:

$$u^* = 1 \quad ; \quad \hat{\boldsymbol{\alpha}}^* = -\frac{\boldsymbol{\lambda}_v}{||\boldsymbol{\lambda}_v||} \tag{30}$$

Finally, embedding u^* and $\hat{\boldsymbol{\alpha}}^*$, the Euler-Lagrange equations for the problem are derived and expressed as follows:

$$\begin{cases} \dot{\mathbf{x}} = \dfrac{\partial H}{\partial \boldsymbol{\lambda}}, \\ \dot{\boldsymbol{\lambda}} = -\dfrac{\partial H}{\partial \mathbf{x}} \end{cases} = \begin{cases} \dot{\mathbf{r}} = \mathbf{v}, \\ \dot{\mathbf{v}} = -\dfrac{\mu}{r^3}\mathbf{r} - \dfrac{T_{\max}}{m}\dfrac{\boldsymbol{\lambda}_v}{\|\boldsymbol{\lambda}_v\|}, \\ \dot{m} = -\dfrac{T_{\max}}{I_{\text{sp}} g_0}, \\ \dot{\boldsymbol{\lambda}}_r = -\dfrac{\partial q(\rho)}{\partial \mathbf{r}} - \dfrac{3\mu}{r^5}(\mathbf{r}\cdot\boldsymbol{\lambda}_v)\mathbf{r} + \dfrac{\mu}{r^3}\boldsymbol{\lambda}_v, \\ \dot{\boldsymbol{\lambda}}_v = -\boldsymbol{\lambda}_r, \\ \dot{\lambda}_m = -\dfrac{\|\boldsymbol{\lambda}_v\| T_{\max}}{m^2}. \end{cases} \tag{31}$$

As the final time t_f of the transfer is an unknown, the last step required to fully define the problem is the establishment of a suitable Transversality Condition. Since the target final state is fixed point in the J2000 equatorial reference frame, the condition simplifies to:

$$H(t_f) - \boldsymbol{\lambda}(tf)\cdot\dot{\boldsymbol{\psi}}(tf) = 0 \longrightarrow H(tf) = 0 \tag{32}$$

In this framework, the Two-Point Boundary Value Problem (TPBVP) is addressed using a shooting method, aiming to determine the unique initial costate vector λ_0 and the final time of transfer t_f such that the propagated state and costate $\boldsymbol{\varphi}([\mathbf{x}_0; \boldsymbol{\lambda}_0], t_0, t_f)$ produce a solution that

minimize the objective function by solving the EL equations while concurrently satisfying the boundary conditions specified in Eq. (24) and the transversality condition through the resolution of the following zero-finding problem:

$$\begin{cases} \mathbf{r}(t_f) - \mathbf{r}_f = 0 \\ \mathbf{v}(t_f) - \mathbf{v}_f = 0 \\ \lambda_m(t_f) = 0 \\ H(t_f) = 0 \end{cases} \tag{33}$$

3.4 Continuous Guidance Solution

In this section, the procedure adopted to solve the zero-finding problem specified in Eq. (33) is outlined. To initialize the solver, the initial costate vector $\boldsymbol{\lambda}_0$ was randomly generated within the interval $[-250; 250]$, while the guess for the final time was set to $t_f = 20\pi$. The pseudo-code of the algorithm employed is reported in Algorithm 1.

Algorithm 1 Zero-Finding Procedure

Require: $\mathbf{x}_0$, t_0, $\boldsymbol{\lambda}_0$, t_f, tol

1: **while** $residuals > tol$ **do**
2: $\quad [\mathbf{x}(t_f), \boldsymbol{\lambda}(t_f)] \leftarrow \boldsymbol{\varphi}([\mathbf{x}_0; \boldsymbol{\lambda}_0], t_0, t_f)$
3: $\quad H(t_f) \leftarrow q(\mathbf{r}(t_f)) + \boldsymbol{\lambda}(t_f) \cdot \mathbf{f}(t_f)$
4: $\quad residuals \leftarrow \begin{bmatrix} \mathbf{r}(t_f) - \mathbf{v}_f \\ \mathbf{v}(t_f) - \mathbf{v}_f \\ \lambda_m(t_f) \\ H(t_f) \end{bmatrix}$
5: $\quad$ **if** $residuals > tol$ **then**
6: $\quad\quad$ Regenerate random $\boldsymbol{\lambda}_0$ and reset t_f
7: $\quad$ **end if**
8: **end while**
9: Return correct $\boldsymbol{\lambda}_0$ and t_f

The solver utilized to address the problem was MATLAB `fsolve` function, employing `levenberg-marquardt` as optimization algorithm.

To enhance the accuracy and robustness of the shooting procedure, as well as to reduce its computational time, the analytic derivatives of the zero-finding problem are provided to the solver. In particular, let $y = [\boldsymbol{\lambda}_0; t_f]$ the set of variables, and $\mathbf{F}$ the Eq. (33), the Jacobian of $\mathbf{F}$ is defined as follows.

$$\frac{\partial \mathbf{F}}{\partial \mathbf{y}} = \begin{bmatrix} \dfrac{\partial \mathbf{r}_f}{\partial \boldsymbol{\lambda}_0} & \dfrac{\partial \mathbf{r}_f}{\partial t_f} \\ \dfrac{\partial \mathbf{v}_f}{\partial \boldsymbol{\lambda}_0} & \dfrac{\partial \mathbf{v}_f}{\partial t_f} \\ \dfrac{\partial \lambda_{m,f}}{\partial \boldsymbol{\lambda}_0} & \dfrac{\partial \lambda_{m,f}}{\partial t_f} \\ \dfrac{\partial H_f}{\partial \boldsymbol{\lambda}_0} & 0 \end{bmatrix}_{8\times 8} \tag{34}$$

By applying the chain rule of derivatives and leveraging the structure of the left-hand side of Eq. (31), the final term can be reformulated as follows:

$$\frac{\partial H_f}{\partial \boldsymbol{\lambda}_0} = \frac{\partial H_f}{\partial \mathbf{x}_f}\frac{\partial \mathbf{x}_f}{\partial \boldsymbol{\lambda}_0} + \frac{\partial H_f}{\partial \boldsymbol{\lambda}_f}\frac{\partial \boldsymbol{\lambda}_f}{\partial \boldsymbol{\lambda}_0} = -\dot{\boldsymbol{\lambda}}_f \frac{\partial \mathbf{x}_f}{\partial \boldsymbol{\lambda}_0} + \dot{\mathbf{x}}_f \frac{\partial \boldsymbol{\lambda}_f}{\partial \boldsymbol{\lambda}_0} \tag{35}$$

Finally, denoting $\mathbf{f}$ the right-hand side of Eq. (31), and computing its state transition matrix $\mathbf{\Phi}(t_i, t_f)$ the final formula for the Jacobian of Eq. (33) is

$$\frac{\partial \mathbf{F}}{\partial \mathbf{y}} = \begin{bmatrix} \mathbf{\Phi}_{\mathbf{r}_f, \boldsymbol{\lambda}_0} & \mathbf{f}_{\mathbf{r}_f} \\ \mathbf{\Phi}_{\mathbf{v}_f, \boldsymbol{\lambda}_0} & \mathbf{f}_{\mathbf{v}_f} \\ \mathbf{\Phi}_{\lambda_m, \boldsymbol{\lambda}_0} & \mathbf{f}_{\lambda_m} \\ -\mathbf{f}_{\boldsymbol{\lambda}_f} \mathbf{\Phi}_{\mathbf{x}_f, \boldsymbol{\lambda}_0} + \mathbf{f}_{\mathbf{x}_f} \mathbf{\Phi}_{\boldsymbol{\lambda}_f, \boldsymbol{\lambda}_0} & 0 \end{bmatrix}_{8\times 8} \tag{36}$$

where the first subscript indicates the rows of the STM considered in the specific submatrix, while the second subscript represents the corresponding columns.

The final optimal solution obtained using the method described above is presented in Table 14, detailing the initial costate vector and the final transfer time, alongside the remaining propellant mass. In Fig. 12 is invece depicted the orbit raising from the initial state to the target point.

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).

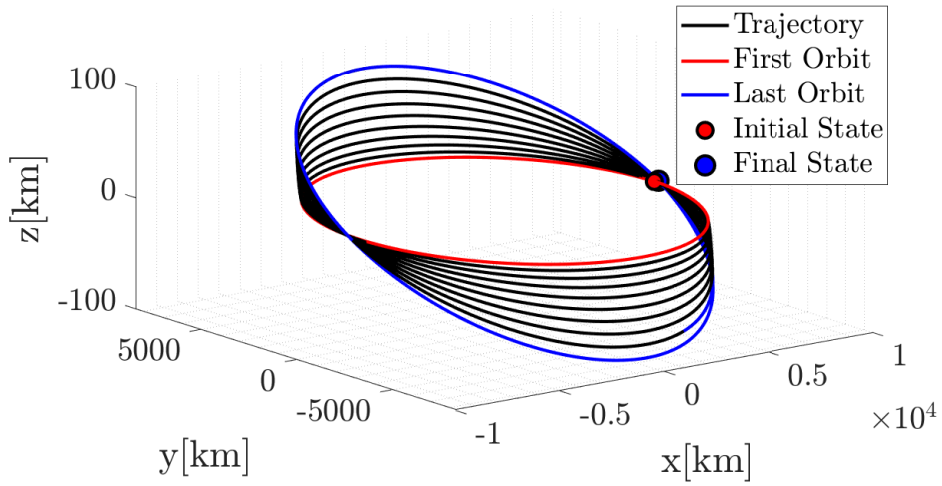

Figure 12: Orbital raising from initial state to target point(@Earth J2000 frame).

Table 15 presents the final errors between the propagated solution and the target values for both position and velocity. As observed, the errors are remarkably low, demonstrating the successful implementation and accuracy of the method.

Error	**Value**	**Units**
$\|\|\mathbf{r}(t_f) - \mathbf{r}_f\|\|$	$5.4043e-09$	km
$\|\|\mathbf{v}(t_f) - \mathbf{v}_f\|\|$	$5.5940e-09$	m/s

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

As a further validation of the solution, the time evolution of the Hamiltonian was computed. According to the theory, for time-independent problems, the Hamiltonian remains constant and, due to the transversality condition in Eq. (32), is equal to zero. Fig. 13 demonstrates this behavior, showing consistency with theoretical expectations, apart from minor numerical errors. Furthermore, in Fig. 13 the time evolution of the primer vector ($\hat{\boldsymbol{\alpha}}$) is shown in the NTW reference frame, defined as:

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

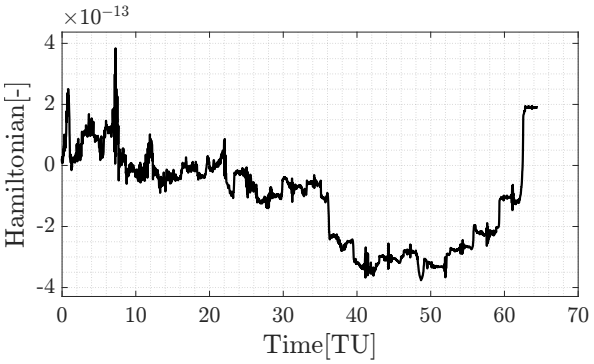

Figure 13: Time evolution of Hamiltonian

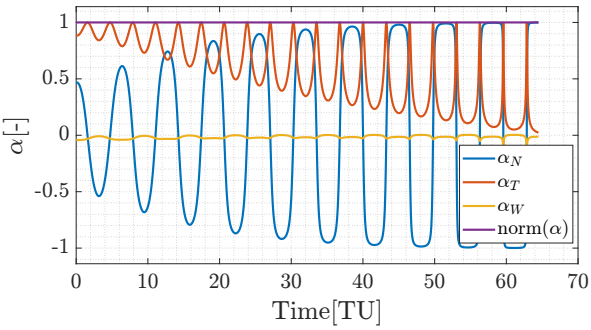

Figure 14: Primer Vector in NTW frame

The observed oscillatory behavior highlights the active control employed to execute the orbit-rasing meaneuver while minimizing the risk of debris impact. The tangential component T, primarily responsible for the orbit raising, is predominantly in the early phases of the trajectory in which, as shown in Fig. 11, the debris density is the highest. The N-component, representing the out-of-plane thrusting, exhibits an oscillatory behavior with a period approximately equal to one orbital revolution, reaching peak amplitude along the line of nodes. This behavior suggests that out-of-plane thrusting is primarily executed at the line of nodes, with positive thrust applied at the ascending node and negative thrust at the descending one. Furthermore, its amplitude increases over time, indicating that during the initial phases of the transfer, a significant portion of the thrust is allocated to orbit-raising maneuvers. Ultimately, the W-component remains consistently close to zero, indicating that radial thrusting is not necessary to achieve the desired final state.

Finally, it is worth noting that, during the implementation of the algorithm, a solution with a final time $t_f \simeq 22\pi$ was occasionally found. Although the solution was feasible, it was sub-optimal as it required an additional orbital period. To prevent the solver from converging to this solution, a check was incorporated into the `while` loop in Algorithm 1.

3.5 Reduced Thrust Analysis

The continuous guidance problem was also solved for a lower thrust level, specifically $T = 2.860\,\text{N}$, corresponding to a scaled value of $T = 3.6970 \times 10^{-4}$. Due to computational efficiency reasons, the numerical continuation technique was adopted, starting from the solution of the problem found in Section 3.4. A vector of $n = 10$ equally spaced thrust levels was generated to iteratively solve the problem. The algorithm for the zero-finding problem remained the same as outlined in Algorithm 1, with the only difference being that the initial guess is not randomly generated but is taken from the previous step. Another slight change was the substitution of the `fsolve` algorithm from `levenberg-marquardt` to `trust-region-dogleg`, which proved to be more effective when the initial guess was close to the solution.
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 ($T_{\max} = 2.860$ N).

Table 17 presents the final errors between the propagated solution and the target values for both position and velocity. As observed also exploiting numerical continuation, the errors remain low demonstrating the successful implementation and accuracy of the proposed strategy.

Error	**Value**	**Units**
$\|\|\mathbf{r}(t_f) - \mathbf{r}_f\|\|$	$5.9519e-09$	km
$\|\|\mathbf{v}(t_f) - \mathbf{v}_f\|\|$	$5.6370e-09$	m/s

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

Also in this case, the solution was validated by computing the Hamiltonian, which, as shown in Fig. 15, remains constant throughout the orbit-raising maneuver, as expected. This behavior confirms the correctness of the implementation and the adherence to the theoretical framework.

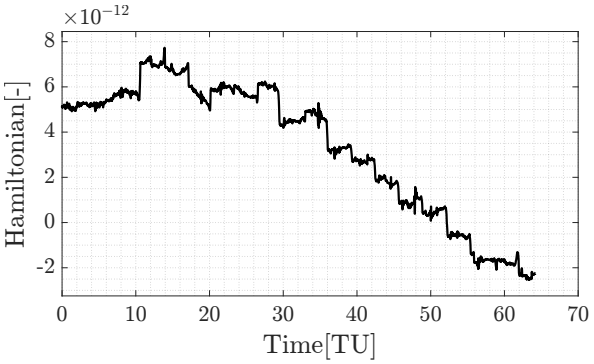

Figure 15: Time evolution of Hamiltonian

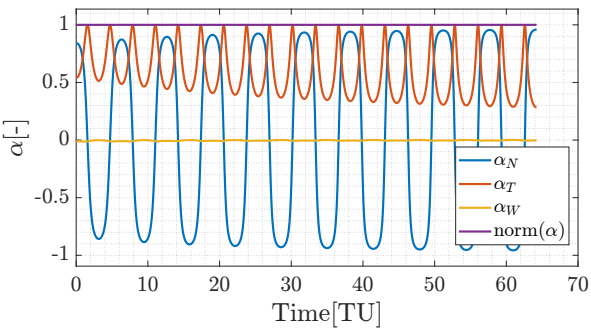

Figure 16: Primer Vector in NTW frame

In Fig. 16, the behavior of the primer vector components under the reduced thrust scenario is illustrated. The components exhibit more regular oscillations, which can be interpreted as an indication of a more efficient utilization of thrust. This improved efficiency may explain why, in the reduced-thrust case, the time required to reach the target orbit is shorter compared to the original scenario, as shown in Table 14 and Table 16.

Appendix

A Analytic Gradients for Simple Shooting

The derivative of the objective function J with respect to the NLP variables: $\nabla J(\mathbf{x}) = \left[\frac{\partial J}{\partial \mathbf{x_0}}, \frac{\partial J}{\partial t_i}, \frac{\partial J}{\partial t_f}\right]^T$, 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:

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

with:

$$\frac{\partial J}{\partial \mathbf{x}_i} = \frac{1}{\sqrt{(v_{x,i} - y_i)^2 + (v_{y,i} + x_i + \mu)^2}} \begin{bmatrix} v_{y,i} + x_i + \mu; \\ y_i - v_{x,i}; \\ v_{x,i} - y_i; \\ v_{y,i} + x_i + \mu \end{bmatrix}$$

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

The time derivatives are calculated as

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

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:

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

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:

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

B 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 of [1].

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:

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

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

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

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

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

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

where:

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

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

References

[1] K. Oshima, F. Topputo, and T. Yanao. "Low-energy transfers to the Moon with long transfer time". In: *Celestial Mechanics and Dynamical Astronomy* 131.4 (2019). DOI: 10.1007/s10569-019-9883-7.

[2] F. 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. URL: https://doi.org/10.1007/s10569-013-9513-8.