SGN – Assignment #1

Gabriele Nuccio, 247457

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

To find the coordinates of the five Lagrange points of the Earth-Moon system the definition of the scalar potential U = $\Omega(x, y, z)$ for the CRTBP has been adopted where: $r_1 = \sqrt{(x+\mu)^2 + y^2 + z^2}$ and $r_2 = \sqrt{(x+\mu-1)^2 + y^2 + z^2}$. The condition defining the libration points $\mathrm{L_i}$ is reported in Eq. (1):

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

Moreover, due to the planar nature of the CRTBP, the z coordinate is equal to zero for all the equilibrium points. Furthermore, the three collinear points ($\mathrm{L_1}$, $\mathrm{L_2}$, $\mathrm{L_3}$) are computed from the condition $\dfrac{\partial U}{\partial x} = 0$ and by considering their y-coordinate equal to zero as they lie on the x-axis of the Earth-Moon rotating frame.

Instead, for the triangular points ($\mathrm{L_4}$, $\mathrm{L_5}$) the conditions of both the partial derivatives of the potential in x and y have to be taken into account.

The solver used to determine the Lagrangian points coordinates is `fsolve` since enables the resolution of systems of equations, more specifically the `OptimalityTolerance` option is set to 10^{-10}, to guarantee convergence to the correct point with the required 10-digit accuracy. For each equilibrium point, initial guesses have been provided, these are chosen among reasonable values to ensure the convergence of the algorithm and considering the relative position of the libration points. In Fig. 1 the coordinates of the different Lagrange points in the Earth-Moon rotating frame are exhibited.

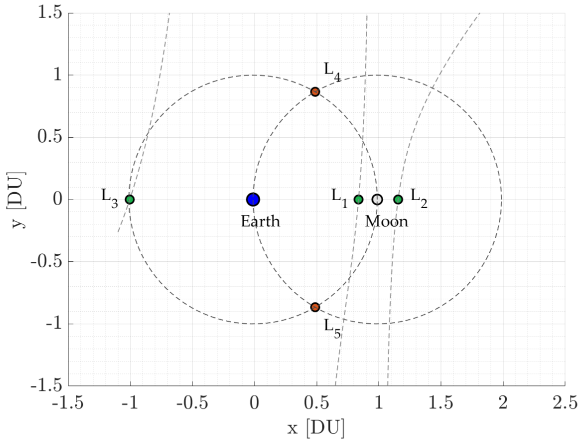

Figure 1: Earth-Moon system Lagrangian Points

Moreover, in Fig. 1, also the partial derivative of the potential $\dfrac{\partial U}{\partial x} = 0$ lines and the unitary circumferences centered in the Earth and in the Moon are represented. These further curves enhance the visibility of the intrinsic features of the Lagrangian points, which are also supported by literature.

Once the coordinates are determined, the Jacobi Constant C corresponding to each of the five libration points can be computed. In this context, the velocity norm (V) is zero, as the equilibrium points, by definition, remain stationary in the Earth-Moon rotating reference 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	0.0000000000	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 Halo Orbit with $C = 3.09$

In this section the provided initial state is intended to be corrected in order to obtain a periodic Halo orbit with a specific associated Jacobi Constant $C = 3.09$. If, in fact, a propagation time is defined (here $\delta = 5$ [TU]), the initial state could be propagated both forward and backward in time to verify if a close periodic orbit is obtained. To assess this also an event function that stops the propagation is integrated. This happens when the condition $y = 0$ is reached. The CRTBP dynamics is described by the equation of motions in Eq. (2):

$$\begin{aligned} \dot{x} &= v_x; \quad \dot{v}_x = \frac{\partial U}{\partial x} + 2v_y \\ \dot{y} &= v_y; \quad \dot{v}_y = \frac{\partial U}{\partial y} - 2v_x \\ \dot{z} &= v_z; \quad \dot{v}_z = \frac{\partial U}{\partial z} \end{aligned} \tag{2}$$

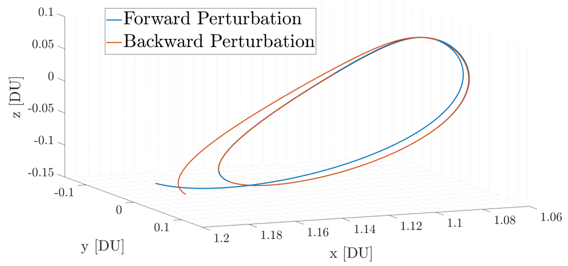

Figure 2: Trajectory propagated from the provided initial state (@EMB Earth-Moon Rotating Frame)

As Fig. 2 shows, the periodicity of the orbit is not guaranteed. To accomplish the request of the exercise and obtain the desired Halo orbit, a theoretical framework has been established. This latter is based on a first-order expansion of the flow with respect to the initial state and final time. Moreover, it incorporates control over the orbit's periodicity and the Jacobi Constant's value through an augmented linear system with an iterative correction process. The Taylor expansion of the flow is presented in Eq. (3):

$$\varphi(\mathbf{x}_0 + \Delta\mathbf{x}, t + \Delta t) = \varphi(\mathbf{x}_0, t) + \frac{\partial \varphi}{\partial \mathbf{x}_0}(\mathbf{x}_0, t_f)\Delta\mathbf{x} + \frac{\partial \varphi}{\partial t}(\mathbf{x}_0, t_f)\Delta t \tag{3}$$

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 method chosen to compute the STM is the variational approach as shown in Eq. (4) which was preferred over the finite differences one. The variational approach, in fact, directly integrates the differential equations governing the STM, ensuring a more precise representation of the system's dynamics. Therefore, the STM is computed as:

$$\begin{cases} \dot{\mathbf{\Phi}} = \dfrac{\partial \mathbf{g}}{\partial \mathbf{x}}(t)\mathbf{\Phi} \\ \mathbf{\Phi}_0 = \mathbf{I}_{6\times6} \end{cases} \tag{4}$$

with $\mathbf{g}$ being the right-hand side of the equations of motion for the CRTBP dynamics, exhibited in Eq. (2). Following the same approach, the Jacobi Constant has been linearized around the initial state. It is worth of notice that, for definition, the Jacobi constant variation in time is null and therefore the linearization in time term.

$$C(\mathbf{x}_0 + \Delta\mathbf{x}) = C(\mathbf{x}_0) + \frac{\partial C}{\partial \mathbf{x}_0}\Delta\mathbf{x} \tag{5}$$

Both in Eq. (3) and in Eq. (5), the left-hand side represents the desired state, namely when $y = 0$, a constraint that represents the perpendicular passage at the xz-plane. Moreover also v_x and v_z have to be null to ensure the aforementioned condition. Finally, for the symmetry of the CRTBP, this means that the computed orbit would be periodic. On the other side, the first terms of the right-hand sides represent the state obtained by propagating the initial state $\mathbf{x}_0$. To compute the necessary corrections, an augmented 7×7 linear system has been constructed. The LHS is represented by the vector of errors $\mathbf{e}$, while the RHS is composed by the augmented matrix $\mathbf{G}$ multiplied by the vector of corrections $\mathbf{d}$. More in detail, $\mathbf{G}$ is composed by the STM at final time (6×6), the derivative in time of the state (7×1) and the gradient of the Jacobi Constant (1×7) while $\mathbf{d}$ includes $\Delta\mathbf{x}$, appended with the time correction Δt.

$$\mathbf{e} = \mathbf{G}\mathbf{d}, \quad \mathbf{G} = \begin{bmatrix} \mathbf{\Phi_f} & \dot{\mathbf{x}}_f \\ \nabla C_{\mathbf{x}}(\mathbf{x_0}) & 0 \end{bmatrix}_{7\times7} \tag{6}$$

Furthermore, only four components of the system are constrained in this problem. As explained before, they are y, v_x, v_z to ensure the periodicity of the orbit and C to ensure the requested value of 3.09. The system is thus reduced by considering only the corresponding rows of the matrix $\mathbf{G}$. Moreover, the initial state at the next iteration has to be corrected only in terms of x_0, z_0, v_{y0}, and t_f. Therefore, only the columns of $\mathbf{G}$ related to these variables are used. These considerations lead to the reduction of the system because it is underdetermined, as only four constraints are imposed. Morover, this latter simplification reduces the computational complexity.

To update the corrected values of the initial state for each iteration the following equation is used:

$$\begin{bmatrix} x_0 \\ z_0 \\ v_{y0} \\ t_f \end{bmatrix}_{k+1} = \begin{bmatrix} x_0 \\ z_0 \\ v_{y0} \\ t_f \end{bmatrix}_{k} + \begin{bmatrix} \Phi_{21} & \Phi_{23} & \Phi_{25} & \dot{y}(t_f) \\ \Phi_{41} & \Phi_{43} & \Phi_{45} & \dot{v}_x(t_f) \\ \Phi_{61} & \Phi_{63} & \Phi_{65} & \dot{v}_z(t_f) \\ \frac{\partial C}{\partial x}(\mathbf{x_0}) & \frac{\partial C}{\partial z}(\mathbf{x_0}) & \frac{\partial C}{\partial v_y}(\mathbf{x_0}) & 0 \end{bmatrix}_{k}^{-1} \begin{bmatrix} 0 - y_f \\ 0 - v_{x,f} \\ 0 - v_{z,f} \\ C_{target} - C_f \end{bmatrix}_{k} \tag{7}$$

where $C_{target} = 3.09$. 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 shown in Table 2.

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 $C = 3.09$

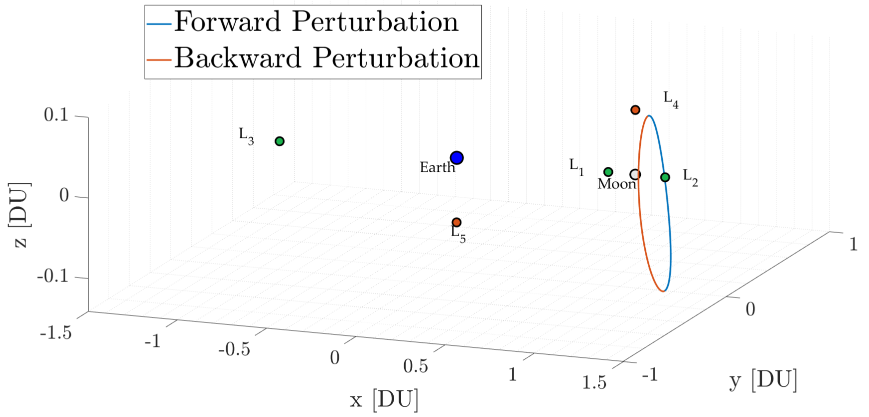

Figure 3: Halo Orbit with $C = 3.09$ (@EMB Earth-Moon Rotating Frame)

The correction scheme successfully converges to a periodic orbit with the target Jacobi constant. The forward and backward propagation from the corrected initial conditions yields the desired halo orbit around L_2.

1.3 Numerical continuation until Halo orbit with $C = 3.04$

This section focuses on determining the correction to the initial state required to obtain a periodic Halo orbit with a Jacobi Constant of 3.04. To ensure rapid convergence of the algorithm, numerical continuation is utilized. Specifically, a family of $n = 11$ is generated, with Jacobi Constants 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 corresponding to the desired C is solved iteratively. In each iteration, the initial guess is updated based on the solution from the previous step, until the final Halo orbit is obtained. The corrected initial state is presented in Table 3.

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

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.

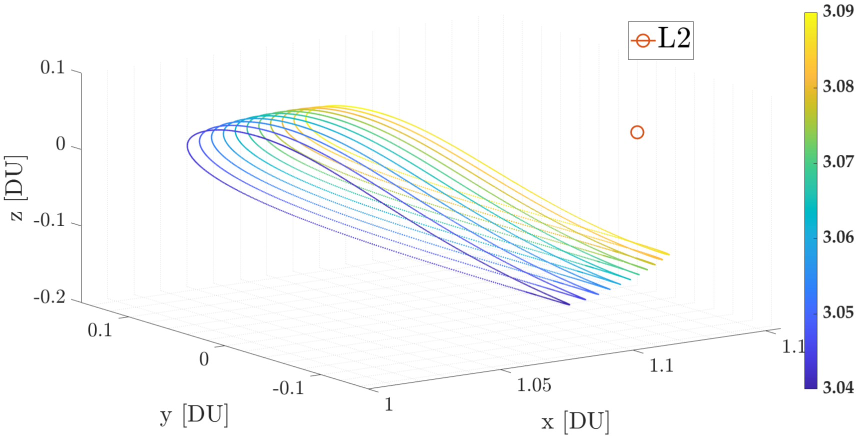

Figure 4: Halo Orbit family from $C = 3.09$ to $C = 3.04$ (@EMB Earth-Moon Rotating Frame)

As the Fig. 4 shows, if C decreases the related mechanical energy and so the dimension of the orbit increases. Furthermore, the allowable region of motion expands. For high C, in fact, motion is tightly constrained near the primary bodies, while lower C allows motion further from these points, enabling larger and more energetic orbits. This explains why halo orbits around L_2 expand with decreasing C.

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 Solution

In the context of the optimization problem for a two-impulse Earth–Moon transfer within the Planar Bicircular Restricted Four-Body Problem (PBRFBP), an initial solution is generated following the procedure outlined in the reference article. [†]. First of all, some essential parameters need to be retrieved, namely the radius and the gravitational parameters to compute the μ parameter as $\mu = \dfrac{\mu_{\text{Moon}}}{\mu_{\text{Earth}} + \mu_{\text{Moon}}}$. The parameters are obtained from the `cspice_bodvrd` function. Given $\{\alpha, \beta, t_i, \delta\}$ and the retrieved parameters, an initial solution is provided. Therefore, the adimensionalized radius r_0 of the initial parking orbit has been computed using the distance unit (DU), starting from Earth's radius and the provided h_i (the same has been done to compute the final parking orbit radius r_f). Let $v_0 = \beta\sqrt{(1-\mu)/r_0}$, the initial state $\mathbf{x}_0(\alpha,\beta) = \{r_{x,0}, r_{y,0}, v_{x,0}, v_{y,0}\}$ is computed as:

$$r_{x,0} = r_0 \cos\alpha - \mu, \quad r_{y,0} = r_0 \sin\alpha, \quad v_{x,0} = -(v_0 - r_0)\sin\alpha, \quad v_{y,0} = (v_0 - r_0)\cos\alpha$$

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) from t_i to $t_f = t_i + \delta$.

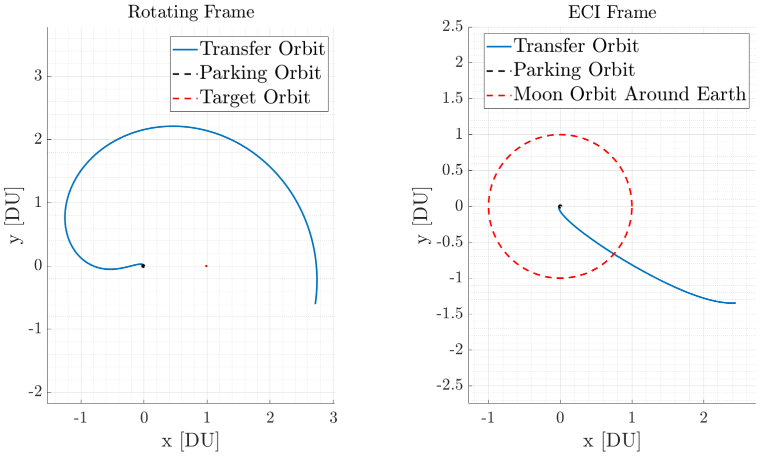

Figure 5: Initial Propagation (@EMB Earth-Moon Rotating Frame) and (@Earth ECI Frame)

In Fig. 5 the trajectory in the Earth-Moon rotating frame and in ECI is reported. The vector of propagated states $\mathbf{x}$ is, in fact, also converted into the Earth-Centered Inertial (ECI) frame, applying a coordinate transformation accounting for the rotation with time of the Earth-Moon

[†] 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.

system, as suggested in the reference article. (Appendix 1, Topputo, 2013). As observed, the first guess solution does not result in an acceptable transfer, as the propagated final state fails to achieve the Moon parking orbit. It is possible to note that the initial state satisfies the initial parking orbit constraints. Therefore, the initial state presented in Table 5 served solely as an initial guess that will require correction in next steps. With this aim, an optimization method will be implemented.

2.2 Simple Shooting

In this section, the simple shooting method is employed to solve the boundary value problem (BVP) for optimizing the desired trajectory. Within the framework of orbit optimization, the spacecraft is required to follow a trajectory that satisfies both the initial and final constraints, defined in terms of position and velocity, while meeting the specified performance objectives. The analysis is performed considering the initial guess presented in Section 2.1 adding the initial and final times. The cost function to be minimized computes the total ΔV as:

$$\Delta V(\mathbf{x_0}, t_i, t_f) = \Delta V_1(\mathbf{x_0}) + \Delta V_2(\varphi(\mathbf{x_0}, t_i, t_f))$$

required for an orbital transfer in the framework of the PBRFBP. Here the BVP is reformulated into an initial value problem (IVP), adjusting the initial conditions. The total ΔV is obtained as the sum of two different impulses that define the velocity required to complete the transfer. The ΔV_1 represents the change of velocity needed to match the spacecraft's velocity with the parking orbit. To define the other impulse, first the equations of motion are integrated from t_i to t_f. ΔV_2 is so the change of velocity required to match the spacecraft's velocity with the target orbit around Moon.

Furthermore, both position and velocity are constrained at initial and final time with the equality constraints. Instead, only one inequality constraint is computed, the one that excludes solutions with a negative flight time between the initial and target orbits ($t_i - t_f < 0$). Both objective function and constraints formulation are presented in the reference article (Section 3.1, Topputo, 2013). This problem is then mapped to an algebraic nonlinear programming (NLP) problem, stated as:

$$\min_{(\mathbf{x}_0, t_i, t_f)} f(\mathbf{x}_0, t_i, t_f) \quad \text{s.t.} \quad \mathbf{c}(\mathbf{x}_0, t_i, t_f) = 0 \quad \text{where:} \quad \mathbf{c} = \begin{cases} \mathbf{c}_i(\mathbf{x}_0) \\ \mathbf{c}_f(\varphi(\mathbf{x}_0, t_i; t_f)) \end{cases} \tag{17}$$

The chosen solver is the `fmincon` `MatLab` function. This latter enables the resolution of constrained problems by giving a initial guess. The adopted algorithm is `active-set` since it can efficiently utilize a good initial guess and so is ideal where the optimization starts close to the solution as in the problem here analyzed. Moreover, both `ConstraintTolerance` and `OptimalityTolerance` are set to 10^{-10}, in order to obtain precise results.

Two distinct scenarios will be evaluated: one without any derivative and one that provides them and takes into account the STM.

In the first case, the solver is applied in its basic form, relying solely on numerical evaluations of the propagated states to iteratively improve the solution without providing the analytical gradients of the objective function and equality constraints to the solver.

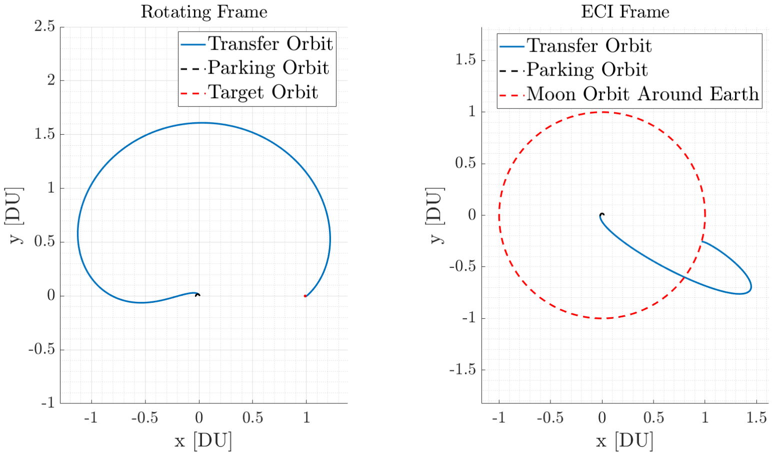

Figure 6: Simple Shooting without gradients trajectory (@EMB Earth-Moon Rotating Frame) and (@Earth ECI Frame)

Fig. 6 shows the obtained transfer trajectory in the rotating frame and in ECI frame.

As requested, in the second case, the solver is equipped with the gradients for both the objective function: $\nabla J(\mathbf{x}) = \left[\frac{\partial J}{\partial \mathbf{x_0}}, \frac{\partial J}{\partial t_i}, \frac{\partial J}{\partial t_f}\right]^T$ and for the equality constraints: $\nabla \mathbf{c}(\mathbf{x})$. By estimating the state transition matrix with the variational approach, the method leverages sensitivity information to improve convergence efficiency and accuracy. Therefore, its dynamics is included in the ODEs system formulation, by means of the Jacobian matrix $\mathbf{A}(t)$ of the equations of motion of the PBRFBP. All the gradients and the Jacobian of the dynamics computed in the problem can be found in Appendix A.

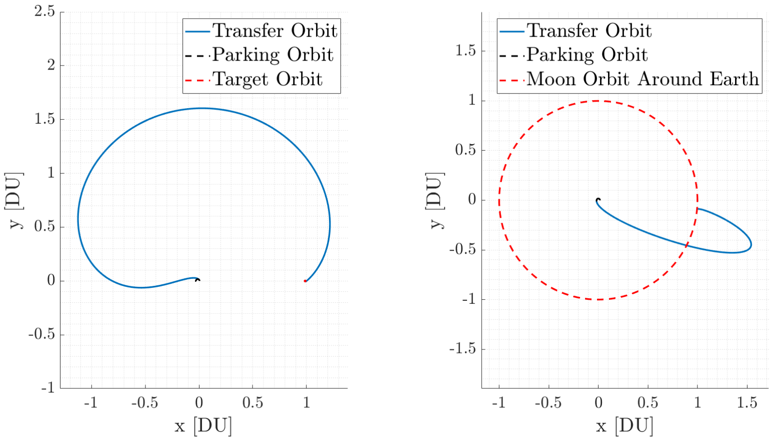

Figure 7: Simple Shooting with gradients trajectory (@EMB Earth-Moon Rotating Frame) and (@Earth ECI Frame)

Fig. 7 shows the obtained transfer trajectory in the second case, both in the rotating frame and in ECI frame. As expected, this method improved the convergence of the solver, decreasing the number of function evaluations of each iteration. Furthermore, an improvement is transfer cost, which decreases from $\Delta V = 4.00755[VU]$ of the first optimization to $\Delta V = 4.00683[VU]$. The optimal variables exiting from the optimization are reported in Table 6.

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	2.026	6.033
True	0.001620	0.010013	-6.290741	8.651415	2.198	6.203

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

In Table 7, the dimentionalized errors at the reference arrival orbit are provided for both simple shooting methods:

Gradients	$\|\boldsymbol{\psi}_r(\mathbf{x}_f)\|$ [m]	$\|\boldsymbol{\psi}_v(\mathbf{x}_f)\|$ [m/s]
False	4.612114×10^{-6}	6.347395×10^{-9}
True	1.004170×10^{-5}	1.465346×10^{-8}

Table 7: Dimentionalized errors at Moon target orbit (Simple Shooting)

The small tolerance specified for the constraints results in very little discrepancies in the final position and velocity for both methods. It can be seen that incorporating the analytical derivatives into `fmincon` does not improve the solution's precision at least in the position components. It is worth to notice that the aim of including the gradients in the solver is just a matter of optimize the transfer cost while optimizing also the convergence time and the number of iterations needed.

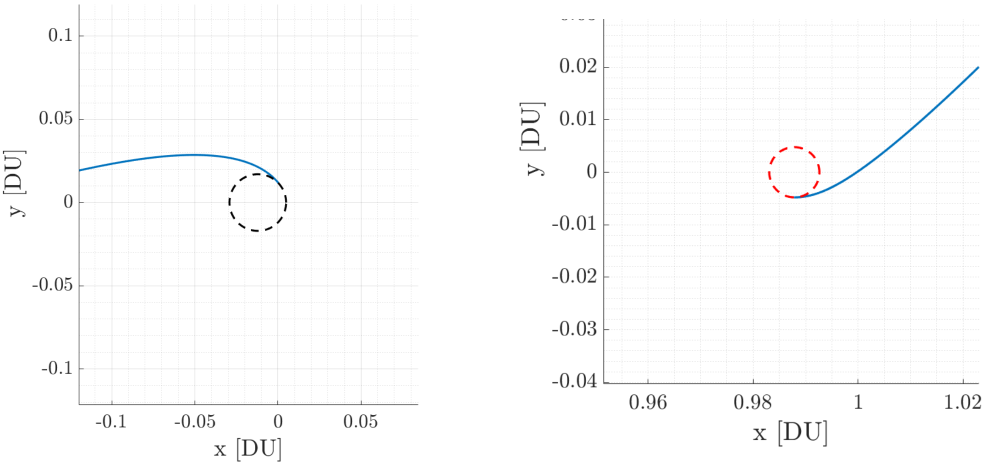

Figure 8: Simple Shooting trajectory details (@EMB Earth-Moon Rotating Frame)

In Fig: 8 is shown a visual representation of the departure from Earth's parking orbit (left) and the arrival to the target orbit around the Moon (right).

2.3 Multiple Shooting

The aim of this section is to describe the multiple shooting method, employed to solve the same problem already solved with the simple shooting approach. In order to implement the multiple shooting method, the optimization state is augmented. The problem is divided into $N = 4$ nodes. This consideration lead the time span to a division into smaller time intervals, which will be solved independently while maintaining continuity across them. Moreover, a reference article was adopted to define the cost function and the constraints that will be further developed. [‡]

[‡] Kenta Oshima, Francesco Topputo, Tomohiro Yanao. 'Low-energy transfers to the Moon with long transfer time', 2019.

Let $\mathbf{x} = \{\mathbf{x_j}, t_1, t_N\}, j = 1, \dots, N$ represent the optimization variables in the problem defined for each of the four nodes. Moreover, t_1 and t_N are the initial and final time instants, respectively; $\mathbf{x_j} = (x_j, y_j, v_{x,j}, v_{y,j})$ represents the state at the j-th node, where $j \in \{1, \dots, N\}$, corresponds to the time step t_j. Furthermore, the multiple shooting approach consists in integrating the dynamics within the $N-1$ intervals ($[t_j, t_{j+1}]$), taking as initial condition $\mathbf{x}_j$.

The objective function is defined as before as

$$J(\mathbf{x}) = \Delta V = \Delta V_1 + \Delta V_2$$

The vector of nonlinear equality constraints of the problem is therefore augmented and includes:

- Initial and final boundary conditions: the position and velocity at the start must match the parking orbit conditions while the position and velocity at the end must match the target orbit features.
- Continuity constraints: for each time interval, the state at the end of one interval must match the state at the start of the next one.

$$\mathbf{c}(\mathbf{x}) = \{\boldsymbol{\zeta}_2, \dots, \boldsymbol{\zeta}_N, \boldsymbol{\psi}_1, \boldsymbol{\psi}_N\} \tag{8}$$

where $\boldsymbol{\zeta}_{j+1}$ represents the defects between the propagated state starting from $\mathbf{x}_j$ and is given by:

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

Moreover, the trajectory optimization is further constrained by ensure that the spacecraft remains at a safe distance from both the Earth and the Moon at each node j, avoiding impacts. Furthermore, even the initial and final times are constrained to maintain consistency in the trajectory duration $\tau = t_1 - t_N < 0$. The vector of nonlinear inequality constraints is:

$$\mathbf{g}(\mathbf{x}) = \{\boldsymbol{\eta}_1, \dots, \boldsymbol{\eta}_N, \tau\} \tag{9}$$

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

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{s.t.} \quad \begin{cases} \mathbf{c}(\mathbf{x}) = 0, \\ g_j(\mathbf{x}) < 0, \quad j = 1, \dots, 2N+1. \end{cases} \tag{10}$$

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. All the analytical expression of the gradients can be consulted in the Appendix B.
Finally, regarding the solver settings, only 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. The `OptimalityTolerance` and the chosen algorithm remain instead the same.

$r_{x,0}$ $[DU]$	$r_{y,0}$ $[DU]$	$v_{x,0}$ $[VU]$	$v_{y,0}$ $[VU]$	t_i $[TU]$	t_f $[TU]$
0.004072	0.005170	-3.247357	10.189489	2.464	6.129

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

Table 8 shows the optimal initial state and the intial and time instants.

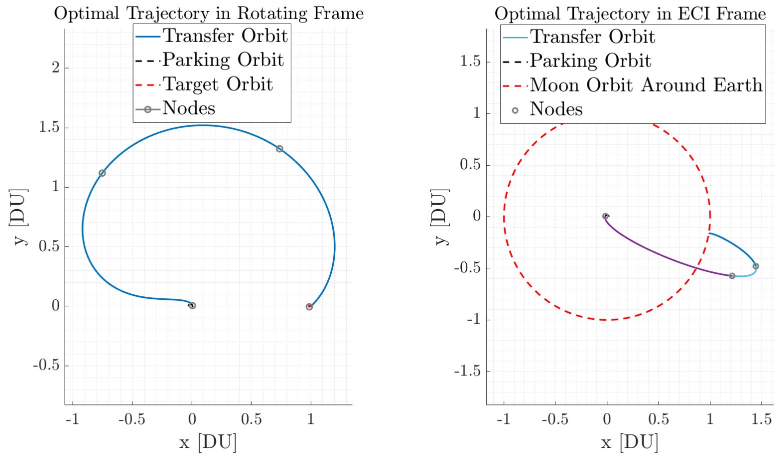

Figure 9: Multiple Shooting trajectory (@EMB Earth-Moon Rotating Frame) and (@Earth ECI Frame)

Fig. 9 represent the optimal transfer in the Earth-Moon rotating frame and in the Earth Centered Inertial frame, respectively. The total adimentionalized cost of the transfer is $\Delta V = 3.99732$ [VU], improving the optimization of the Simple Shooting. This outcome, as expected, highlights the superior accuracy of multiple shooting methods when addressing nonlinear dynamics and extended time of flights. Moreover in Table 9, the residuals at the Moon target orbit ensure are smaller than the one obtained with simple shooting shown in Table 7. This result exhibit higher accuracy with multiple shooting method, at least with respect to the final orbit. However, the higher tolerance of the constraints allows higher errors for the internal nodes of the problem. More accurate outputs can be found by decreasing the tolerance on the constraints. However, this would require higher computational effort for the solver and thus this option is discarded.

$\lvert\boldsymbol{\psi}_r(\mathbf{x}_f)\rvert$ [m]	$\lvert\boldsymbol{\psi}_v(\mathbf{x}_f)\rvert$ [m/s]
4.058976×10^{-8}	$-5.882051 \times 10^{-12}$

Table 9: Dimentionalized errors at Moon target orbit (Multiple Shooting)

2.4 N-Body Propagation

From the solution of the simple shooting method with gradients implemented, reported in Table 6, a 3D N-body propagation in the Earth-Moon-Sun (EMB) system is performed trasforming the solution $\{\mathbf{x}_i, t_i, t_f\}$ in ECI frame and into physical units. The RHS of the N-body propagator can be expressed as:

$$\frac{d}{dt}\begin{bmatrix}\mathbf{r}\\ \mathbf{v}\end{bmatrix} = \begin{bmatrix}\mathbf{v}\\ \mathbf{a}\end{bmatrix}$$

where $\mathbf{a}$ is the total gravitational acceleration, given by $\mathbf{a} = \mathbf{a}_0 + \sum_{i\neq 0}\mathbf{a}_i$
More in detail, $\mathbf{a}_0 = -\frac{\mu_{Earth}\mathbf{r}}{r^3}$ is the gravitational acceleration due to the central body. The rest is the sum of each contribution coming from the i-th perturbing body:

$$\mathbf{a}_i = -\frac{\mu_i}{|\mathbf{r} - \boldsymbol{\rho}_i|^3}\left(\mathbf{r} + \boldsymbol{\rho}_i f(q)\right)$$

where: μ_i is the gravitational parameter of the perturbing body; $\boldsymbol{\rho}_i$ is the position vector of the perturbing body with respect to the center; $f(q)$ is instead a perturbation function defined as:

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

This formulation incorporates the direct gravitational influence of Earth along with additional perturbations from other celestial bodies, providing a more precise representation of the object's motion within a multi-body gravitational framework.

To perform the propagation the initial epoch has to be adjusted to match the data available in the provided kernels, while ensuring compatibility with the initial state conditions derived from Ex. 2.2 b). This is done by aligning the Moon-EMB-Sun angle to the initial angle at the provided epoch.

The initial angle is thus computed as $\theta_0 = (\omega_s \cdot t_i)$, where ω_s is the scaled angular velocity of the Sun given in Table 4. A zero-finding problem is then formulated. First, the angle between Sun and Moon with respect to the EMB is computed. This is perfomed in approximately one Moon revolution period around Earth from the provided starting epoch.

Then, a solver is adopted to find the solution within the aforementioned time span such that the computed angle equals the target angle. The chosen solver is `fzero` since enables the resolution of one dimension zero-finding problem and is used without any other `options` specification.

Afterwards, the final epoch is computed adding the time of flight solution found in the Ex. 2.2 b) to the found initial epoch. The initial state in ECI is instead computed with the rotation already exhibited in Ex. 2.1, considering the third component of both position and velocity equal to zero.

Symbol	**Calendar epoch** (UTC)
t_i	2024-OCT-12-11:40:24.6073
t_f	2024-OCT-29-21:05:10.7673

$r_{x,0}$ $[km]$	$r_{y,0}$ $[km]$	$r_{z,0}$ $[km]$
-6223.6285	2026.1445	0.0

$v_{x,0}$ $[km/s]$	$v_{y,0}$ $[km/s]$	$v_{z,0}$ $[km/s]$
-3.39800	-10.43750	0.0

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

The propagation is performed using `ode113` with the right-hand side evaluated by the function explained before, which models the contribute of gravitational acceleration from N celestial bodies for accurate trajectory prediction. Moreover, the propagation is performed from the computed t_i to t_f.

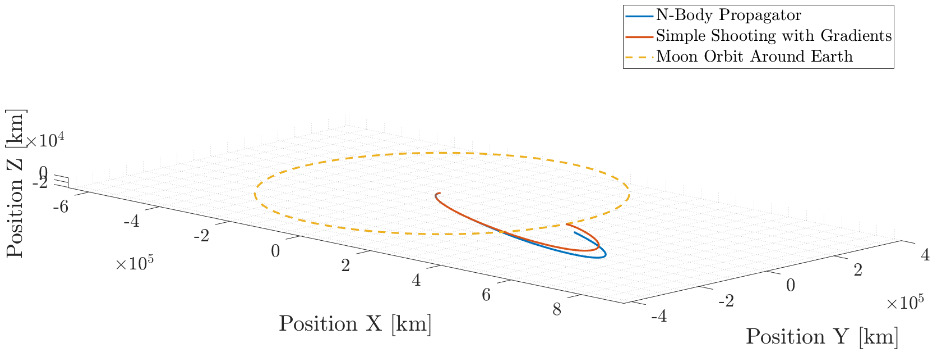

Figure 10: N-Body propagation and Simple Shooting Trajectories (@Earth ECI Frame)

As illustrated in Fig. 10, the trajectory computed using the N-body propagator does not intersect the Moon's orbit around the Earth. This discrepancy arises due to the Moon being represented using a two-dimensional state, whereas the evaluated problem employs a three-dimensional propagation framework. The latter approach allows the initially zero components of z and v_z to grow in magnitude along the transfer orbit since now an additional gravitational accelerations is taken into account. Consequently, as depicted in the detailed view, the transfer orbit deviates from the initial Earth-Centered Inertial (ECI) reference plane.

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)

3.1 Debris Spatial Density

The evolution of the space debris spatial density as a function of orbital altitude is showcased in Fig. 11.

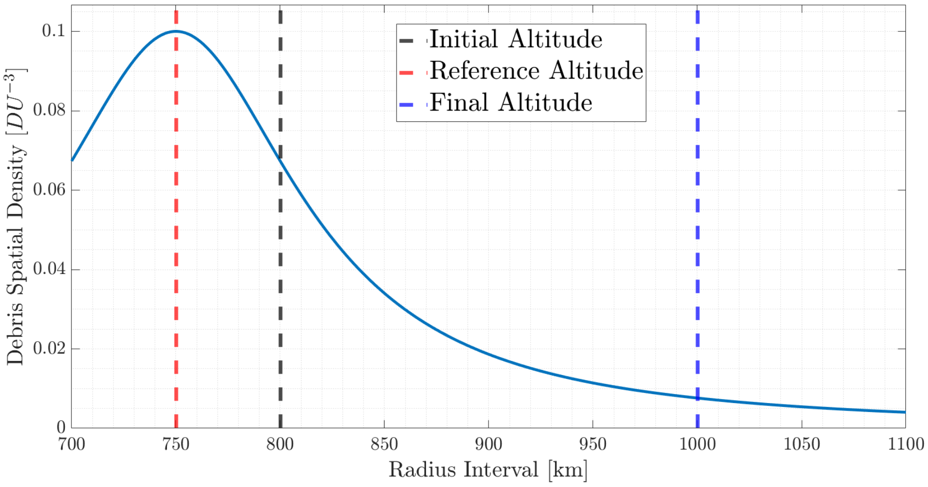

Figure 11: Debris Spatial Density vs Orbital Altitude

The peak in debris density is observed at an altitude 50 km below the initial orbit, corresponding to the reference radius used to define the debris flux. Across the altitude range the population of resident space objects demonstrates a declining trend after the peak.
Knowing that the intial and final state are located in the x-axis of the equatorial J2000 reference frame and that the rotation of angle Δi is done around the same axis, the initial and target states are computed as follows:

$$\mathbf{x}_i = \begin{pmatrix} R_e + h_i \\ 0 \\ 0 \\ 0 \\ \sqrt{\dfrac{\mu}{R_e + h_i}} \\ 0 \end{pmatrix}, \quad \mathbf{x}_f = \begin{pmatrix} R_e + h_f \\ 0 \\ 0 \\ 0 \\ \sqrt{\dfrac{\mu}{R_e + h_f}} \cos \Delta i \\ \sqrt{\dfrac{\mu}{R_e + h_f}} \sin \Delta i \end{pmatrix} \tag{11}$$

The results are reported in Table 12

[§] 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.

$r_{x,i}$ $[km]$	$r_{y,i}$ $[km]$	$r_{z,i}$ $[km]$	$v_{x,i}$ $[km/s]$	$v_{y,i}$ $[km/s]$	$v_{z,i}$ $[km/s]$
7178.136600	0.000000	0.000000	0.000000	7.45183148	0.000000

$r_{x,f}$ $[km]$	$r_{y,f}$ $[km]$	$r_{z,f}$ $[km]$	$v_{x,f}$ $[km/s]$	$v_{y,f}$ $[km/s]$	$v_{z,f}$ $[km/s]$
7378.136600	0.000000	0.000000	0.000000	7.34950907	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, and imposing the adimensional Earth gravitational parameter to $\mu = 1$ as requested, first the time and velocity units are computed as:

$$TU = \sqrt{\frac{DU^3}{\mu_{dim}}}, \quad VU = \frac{DU}{TU} \tag{12}$$

TU [s]	VU [km/s]	h_i [DU]	h_f [DU]	R_e [DU]	μ $[DU^3/TU^2]$
963.2715	7.45183	0.11145	0.13931	0.88855	1

ρ_0 [DU]	m_0 [MU]	$T_{\max}$ $\left[\frac{MU\ DU}{TU^2 \cdot 1000}\right]$	I_{sp} [TU]	g_0 $\left[\frac{DU}{TU^2 \cdot 1000}\right]$
0.99303	1	3.8780×10^{-4}	3.23896	1.2681

Table 13: Adimensionalized Parameters and Constants

In Eq. (13), the adimensionalized initial and target states are reported:

$$\mathbf{x}_i = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} [DU,\ DU/TU], \quad \mathbf{x}_f = \begin{pmatrix} 1.02786 \\ 0 \\ 0 \\ 0 \\ 0.98627 \\ 0.01291 \end{pmatrix} [DU,\ DU/TU] \tag{13}$$

3.3 Formulation of the PMP

In this section, the problem is formulated based on the Pontryagin Maximum Principle (PMP). 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:

$$l(\mathbf{x}, \mathbf{u}, t) = q(\rho(t)) \tag{14}$$

The problem is therefore formulated as follows:

$$\min_{(\hat{\boldsymbol{\alpha}}, u) \in \Omega} \int_{t_0}^{t_f} l(\mathbf{x}, \mathbf{u}, t)\, dt \quad \text{s.t.} \quad \begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \hat{\boldsymbol{\alpha}}, t) \\ \mathbf{x}(t_0) = \mathbf{x}_0 \\ \mathbf{r}(t_f) = \mathbf{r}_f \\ \mathbf{v}(t_f) = \mathbf{v}_f \\ \boldsymbol{\lambda}_m(t_f) = 0 \end{cases} \tag{15}$$

Where Ω is the set of admissible control actions: in particular $\hat{\boldsymbol{\alpha}}$ is the thrust direction, and $u \in (0, 1)$ is the throttle factor.

The Hamiltonian of the problem is defined as:

$$H = l(\mathbf{x}, \mathbf{u}, t) + \boldsymbol{\lambda}^T \mathbf{f} = q(\rho(t)) + \big(\boldsymbol{\lambda}_r,\ \boldsymbol{\lambda}_v,\ \lambda_m\big) \begin{pmatrix} \mathbf{v} \\ -\dfrac{\mu}{r^3}\mathbf{r} + u\dfrac{T_{\max}}{m}\hat{\boldsymbol{\alpha}} \\ -u\dfrac{T_{\max}}{I_{\text{sp}}g_0} \end{pmatrix} \tag{16}$$

By applying PMP, which determines the optimal control policy among admissible controls by minimizing the Hamiltonian, the optimal control is expressed in terms of control magnitude (u^*) and thrust direction, $\hat{\boldsymbol{\alpha}}^*$. This latter is defined as follows, ensuring alignment with the negative gradient of the velocity factor.

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

Consequently, the Euler-Lagrange Equations are limitate to the state and costate dynamics. Adding the equation for minimizing the Hamiltonian, the equations are defined as:

$$\begin{cases} \dot{\mathbf{x}} = \dfrac{\partial H}{\partial \boldsymbol{\lambda}} \\ \dot{\boldsymbol{\lambda}} = -\dfrac{\partial H}{\partial \mathbf{x}} \\ (u^*, \hat{\boldsymbol{\alpha}}^*) = \arg\min_{\mathbf{u}\in\Omega} H(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{u}) \end{cases} = \begin{cases} \dot{\mathbf{r}} = \mathbf{v} \\ \dot{\mathbf{v}} = -\dfrac{\mu}{r^3}\mathbf{r} - u\dfrac{T_{\max}}{m}\dfrac{\boldsymbol{\lambda}_v}{\|\boldsymbol{\lambda}_v\|} \\ \dot{m} = -u\dfrac{T_{\max}}{I_{\text{sp}}g_0} \\ \dot{\lambda}_r = -\dfrac{\partial q(\rho(t))}{\partial \mathbf{r}} - \dfrac{3\mu}{r^5}(\mathbf{r}\cdot\boldsymbol{\lambda}_v)\mathbf{r} + \dfrac{\mu}{r^3}\boldsymbol{\lambda}_v \\ \dot{\boldsymbol{\lambda}}_v = -\boldsymbol{\lambda}_r \\ \dot{\lambda}_m = -u\dfrac{\lambda_v T_{\max}}{m^2} \end{cases}$$

Typically, the control u is determined based on an optimal thrust policy by evaluating the sign of the switching function, defined as

$$S = -\frac{\lambda_v}{m}I_{\text{sp}}g_0 - \lambda_m.$$

In this scenario, however, since $\dot{\lambda}_m < 0$ and $\lambda_m(t_f) = 0$, it follows that $\lambda_m > 0$ and so $S < 0$ for all $t \in [t_0, t_f]$. As a result, the control can be consistently assumed to remain active, with a constant thrust throttle factor $u = 1$. This consideration prompted the treatment of the problem as a time-optimal optimization.
Additionally, to find t_f, the proper transversality condition is reported. Since the target is a fixed point in space and so is not moving in time, $\dot{\boldsymbol{\psi}}(t_f) = 0$.

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

In order 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 for which the boundary conditions (Eq. (15)) and the transversality condition are met.

3.4 Continuous Guidance Solution

The purpose of this section is to determine the optimal trajectory of the spacecraft by solving for the initial costates ($\boldsymbol{\lambda}_\mathbf{0}$) and the final time ($t_f$), while ensuring compliance with the boundary conditions at the initial and final states and with the transversality condition at the final time. The objective function aims to identify the optimal values of these variables to minimize the integral of debris density along the trajectory during the flight time. This is accomplished

through the formulation of a zero-finding problem.
Moreover, an iterative approach is employed, relying on random guesses to solve the proposed Initial Value Problem, as opposed to a conventional shooting method. The latter typically requires a good initial guess to ensure convergence to the correct solution. Poor initial estimates often lead the solver to diverge or settle on incorrect local solutions. The use of random guesses, instead, facilitates a broader exploration of the solution space, reducing the risk of being trapped in local minima. This is particularly advantageous for problems where solutions are sparse or confined to narrow regions.
The initial guesses for $\boldsymbol{\lambda_0}$ and t_f are generated randomly, with $\boldsymbol{\lambda}_{0,r}$ and $\boldsymbol{\lambda}_{0,v} \in [-250, +250]$, while $\lambda_{0,m} \in [0, +250]$ as mass cannot be negative. Finally, t_f is set to 20π. Once the initial guesses are provided, the zero-finding problem, as outlined in Algorithm 1, is solved using MATLAB's `fsolve` solver. Furthermore, the Levenberg-Marquardt algorithm was selected for the optimization. This algorithm is, in fact, better suited for cases where the initial guess may be relatively far from the optimal solution and thus better suited for an approach based on random guesses.
To enhance both the efficiency and accuracy of convergence, as well as to reduce its computational time, the solver was supplied with the analytical derivatives of the boundary conditions and the Hamiltonian. Consider the set of variables $\mathbf{y} = [\boldsymbol{\lambda}_0; t_f]$ and let $\mathbf{B}$ represent the combination of boundary and transversality conditions. The Jacobian matrix of $\mathbf{B}$ with respect to $\mathbf{y}$ is defined as:

$$\frac{\partial \mathbf{B}}{\partial \mathbf{y}} = \begin{bmatrix} \dfrac{\partial \mathbf{r}_f}{\partial \boldsymbol{\lambda}_0} & \dfrac{\partial \mathbf{r}_f}{\partial t_f} \\ \dfrac{\partial \mathbf{v}_f}{\partial \boldsymbol{\lambda}_0} & \dfrac{\partial \mathbf{v}_f}{\partial t_f} \\ \dfrac{\partial \lambda_{m,f}}{\partial \boldsymbol{\lambda}_0} & \dfrac{\partial \lambda_{m,f}}{\partial t_f} \\ \dfrac{\partial H_f}{\partial \boldsymbol{\lambda}_0} & 0 \end{bmatrix}_{8\times 8} \tag{17}$$

Notably, the derivative of the Hamiltonian with respect to time is zero. Furthermore, applying the chain rule to compute the derivative of the Hamiltonian with respect to the initial costate yields:

$$\frac{\partial H_f}{\partial \boldsymbol{\lambda}_0} = -\dot{\boldsymbol{\lambda}}_f \frac{\partial \mathbf{x}_f}{\partial \boldsymbol{\lambda}_0} + \dot{\mathbf{x}}_f \frac{\partial \boldsymbol{\lambda}_f}{\partial \boldsymbol{\lambda}_0}. \tag{18}$$

Finally, denoting by $\mathbf{f}$ the right-hand side of the Euler-Lagrange equations and using the state transition matrix $\boldsymbol{\Phi}(t_i, t_f)$ derived through the variational approach, the Jacobian matrix can be expressed as:

$$\frac{\partial \mathbf{B}}{\partial \mathbf{y}} = \begin{bmatrix} \boldsymbol{\Phi}_{\mathbf{r}_f, \boldsymbol{\lambda}_0} & \mathbf{f}_{\mathbf{r}_f} \\ \boldsymbol{\Phi}_{\mathbf{v}_f, \boldsymbol{\lambda}_0} & \mathbf{f}_{\mathbf{v}_f} \\ \boldsymbol{\Phi}_{\lambda_m, \boldsymbol{\lambda}_0} & \mathbf{f}_{\lambda_m} \\ -\mathbf{f}_{\boldsymbol{\lambda}_f} \boldsymbol{\Phi}_{\mathbf{x}_f, \boldsymbol{\lambda}_0} + \mathbf{f}_{\mathbf{x}_f} \boldsymbol{\Phi}_{\boldsymbol{\lambda}_f, \boldsymbol{\lambda}_0} & 0 \end{bmatrix}_{8\times 8}. \tag{19}$$

Here, the subscript of the state transition matrix (STM) $\boldsymbol{\Phi}$ denotes the rows and columns involved in the respective submatrices. The STM, $\boldsymbol{\Phi}(t_i, t_f)$, is defined as:

$$\boldsymbol{\Phi}(t_i, t_f) = \begin{bmatrix} \dfrac{\partial \mathbf{x}_f}{\partial \mathbf{x}_0} & \dfrac{\partial \mathbf{x}_f}{\partial \boldsymbol{\lambda}_0} \\ \\ \dfrac{\partial \boldsymbol{\lambda}_f}{\partial \mathbf{x}_0} & \dfrac{\partial \boldsymbol{\lambda}_f}{\partial \boldsymbol{\lambda}_0} \end{bmatrix}_{14\times 14}$$

Each block of the STM contributes to the submatrices of the Jacobian.
This formulation ensures that the Jacobian is computed analytically, improving the performance and accuracy of the numerical shooting method.

Algorithm 1 Cost Function for Initial Costate and Final Time Estimation

```
**Inputs**: $\boldsymbol{\lambda_0}, t_f$: Initial guess,
$\mathbf{x}_{\text{target}}$: Target final conditions,
$\mathbf{x}_0$: Initial state vector.
**Outputs**: Optimal solution: $\mathbf{x}* = [\boldsymbol{\lambda}_0^*; t_f^*]$
Initialize state vector and costate $[\mathbf{x}_0; \boldsymbol{\lambda_0}]$
Initialize state transition matrix $\boldsymbol{\Phi}_0$ as an identity matrix $\mathbf{I}_{14\times 14}$
**repeat**
**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 STM $\boldsymbol{\Phi}_f$ from the propagation results
**Compute Hamiltonian:**
Evaluate the Hamiltonian $H$ at final time
**Compute the Derivatives:**
Use the final state transition matrix $\boldsymbol{\Phi}_f$ and final state to compute the gradient $\nabla \mathbf{f}$.
**Compute Error Vector:**
**err** := $[\Delta \mathbf{x}_f; \Delta \lambda_{m,f}; \Delta H(t_f)]$
Update $\boldsymbol{\lambda_0}$ and $t_f$ using fsolve
**until** All errors are below the tolernce
**Return** $\mathbf{x}* = [\boldsymbol{\lambda}_0^*; t_f^*]$
```

Once a feasible initial guess is found, it is propagated up to the the final state by integrating the equations of motion.
The optimal solution in terms of initial costate vector and final time is reported in Table 14.

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).

In Fig. 12 is shown the orbit raising trajectory from the initial to the target point.

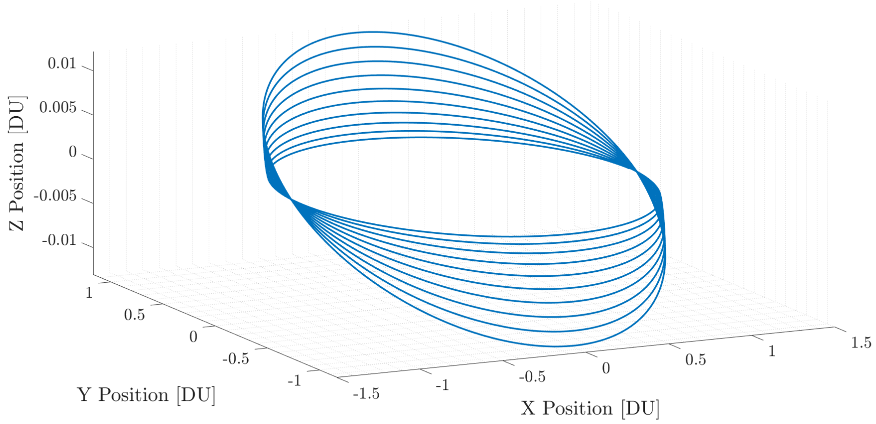

Figure 12: Orbit Raising Trajectory for $T_max = 3.000$ N (@Earth J2000)

It is worth of notice that, during the solution of the zero-finding problem, an alternative transfer is occasionally found, corresponding to a final time of $t_f \approx 22\pi$. However, this solution is discarded as it required an additional orbital period to complete the transfer and thus result in a feasible solution but further in time than the one reported in Table 14.

Error	**Value**	**Units**
$\|\|\mathbf{r}(t_f) - \mathbf{r}_f\|\|$	9.93727e-10	km
$\|\|\mathbf{v}(t_f) - \mathbf{v}_f\|\|$	8.99238e-10	m/s

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

As previously noted, the transversality condition requires the Hamiltonian to be zero at the final time. To validate the obtained solution, the time evolution of the Hamiltonian was computed over the entire transfer and is presented in Fig. 13. As shown, the Hamiltonian remains constant throughout the transfer, which is consistent with the behavior expected in not time-dependent problems and time-optimal solutions. This constancy indicates that no further improvements can be achieved without violating the system's constraints, thereby confirming the global optimality of the solution.

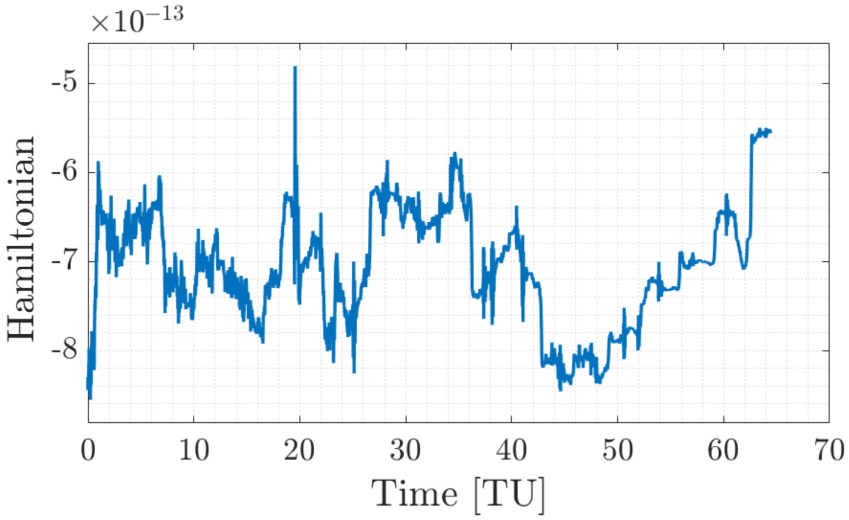

Figure 13: Hamiltonian over time

Finally, as requested, the time evolution of the primer vector in the NTW frame was computed. First, the transformation from ECI to NTW is computed as:

$$\hat{\boldsymbol{\alpha}}_{\mathrm{NTW}} = \mathbf{R} \cdot \hat{\boldsymbol{\alpha}}_{\mathrm{ECI}}, \quad \mathbf{R} = [\hat{\mathbf{N}}, \hat{\mathbf{T}}, \hat{\mathbf{W}}] \tag{20}$$

where:

$$\hat{\mathbf{N}} = \frac{\mathbf{r}_{\mathrm{ECI}} \times \mathbf{v}_{\mathrm{ECI}}}{\|\mathbf{r}_{\mathrm{ECI}} \times \mathbf{v}_{\mathrm{ECI}}\|}, \quad \hat{\mathbf{T}} = \frac{\mathbf{v}_{\mathrm{ECI}}}{\|\mathbf{v}_{\mathrm{ECI}}\|}, \quad \hat{\mathbf{W}} = \hat{\mathbf{N}} \times \hat{\mathbf{T}}. \tag{21}$$

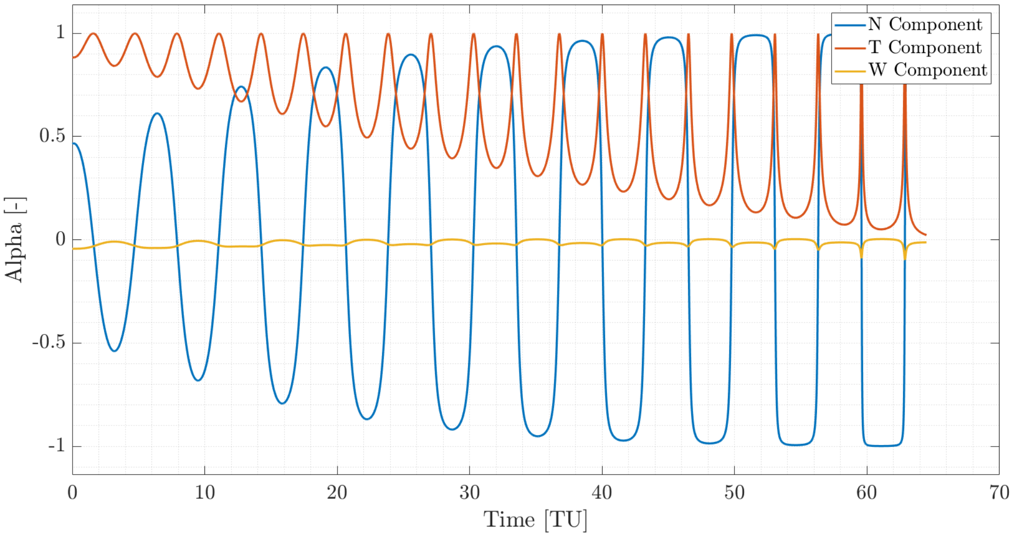

Figure 14: Prime Vector Time Evolution for T = 3.000 (NTW frame)

The observed oscillatory behavior highlights the active control employed to minimize the risk of debris impact while optimizing flight time. The consistently positive values of the T-component indicate that thrust is predominantly tangential, particularly in the first part, aligning with the mission objective: continuous low-thrust orbit raising. This approach prioritizes a significant change in orbital energy, corresponding to $\Delta a = 200\,\mathrm{km}$, over a smaller inclination change of $\Delta i = 0.75°$.

In contrast, the N-component exhibits significant oscillations, almost in correspondence with the orbital period, gradually converging toward amplitudes of ± 1. This oscillatory nature reflects the fact that out-of-plane thrust is applied with the maximum amplitude in the line of nodes in ordenr to increase the orbit inclination. This consideration is enforced by the fact that the RAAN remains the same during all the transfer. Furthermore, the N-component assumes positive amplitudes in correspondence of the ascending node while negative ones at the descending node.

Finally, the W-component remains consistently near zero throughout the maneuver. This indicates that thrusting in this direction is not required, as the maneuver relies in changing the semi-major axis and the inclination.

3.5 Reduced Thrust Level Analysis

In this final analysis, the continuous guidance problem is solved for a reduced maximum thrust level of $T_{\max} = 2.860\,\mathrm{N}$. The procedure for determining the optimal initial costate vector, $\boldsymbol{\lambda}_0$, and the final time, t_f, follows the same approach outlined in Algorithm 1. To improve the convergence efficiency, numerical continuation is incorporated into the process. Specifically, a vector of equally spaced thrust levels ranging from 3.000 N to 2.860 N is defined, and the zero-finding problem is solved iteratively. At each step, the solution obtained from the previous thrust level is used as the initial guess for the subsequent iteration. This iterative approach ensures a smooth convergence to the final solution. Furthermore, here the `trust-region-dogleg` algorithm was selected for the optimization. This algorithm is, in fact, better suited for cases where the initial guess is relatively close to the solution and thus is the optimal one for numerical continuation application such as the one presented here.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).

Error	**Value**	**Units**
$\|\|\mathbf{r}(t_f) - \mathbf{r}_f\|\|$	1.04297e-08	km
$\|\|\mathbf{v}(t_f) - \mathbf{v}_f\|\|$	1.03733e-08	m/s

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

A key observation, when comparing the results with the higher thrust case, is the slight reduction in transfer time. In a standard time-optimal problem, one would typically expect the opposite: higher thrust availability generally enables more intense and efficient maneuvers, leading to a shorter flight time .
However, in the lower thrust case, the spacecraft executes the orbit-raising maneuver more gradually, allowing for more controlled thrusting phases. Thus, despite the lower available thrust, the shorter time of flight is explained.

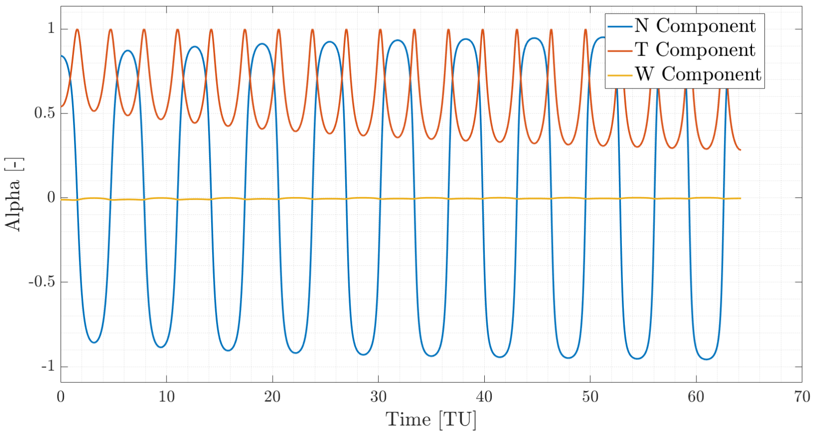

Figure 15: Prime Vector Time Evolution for T = 2.860 N (NTW frame)

As the Fig. 15 shows, the primer vector presents higher amplitudes oscillations with respect to the Fig. 14. All the components reveal more consistent amplitudes throughout all the trajectory, indicating a more efficient thrusting in the low-thrust case and thus upholding the considerations done before. This efficient use of thrust minimizes the time needed to achieve the desired changes in the orbital plane and energy, resulting in a shorter overall flight duration.

Appendix

A Appendix: 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 essential for the computational efficiency and accuracy of the optimization processes. Their explicit expressions are provided here.

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:

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

While 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

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

As for the constraints:

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

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

Obtaining:

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

where $\mathbf{f}$ is the right-hand side of the equations of motion.

The STM $\mathbf{\Phi}$ is propagated with the variational approach such that:

$$\begin{cases} \dot{\mathbf{\Phi}}(t_0, t) = \mathbf{A}(t)\mathbf{\Phi}(t_0, t) \\ \mathbf{\Phi}0 = \mathbf{I}_{4\times4} \end{cases}$$

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 follows:

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

B Appendix: 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.

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:

$$P_1 = \frac{1}{\sqrt{(v_{x,1} - y_1)^2 + (v_{y,1} + x_1 + \mu)^2}} \left[v_{y,1} + x_1 + \mu;\ y_1 - v_{x,1};\ v_{x,1} - y_1;\ v_{y,1} + x_1 + \mu \right],$$

$$P_N = \frac{1}{\sqrt{(v_{x,N} - y_N)^2 + (v_{y,N} + x_N + \mu - 1)^2}} \left[v_{y,N} + x_N + \mu - 1;\ y_N - v_{x,N};\ v_{x,N} - y_N;\ v_{y,N} + x_N + \mu - 1 \right]$$

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:

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

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

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

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

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:

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

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