SGN – Assignment #1

Mina Baniamein, 232521

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 Point 1

In the CR3BP the equation of motion can be written as a function of the derivative of potential U as follows:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, t) = \begin{cases} \dot{x} = v_x \\ \dot{y} = v_y \\ \dot{z} = v_z \end{cases} \tag{1}$$

with:

$$\begin{aligned} \dot{x} &= \frac{\partial U}{\partial x} + 2y \\ \dot{y} &= \frac{\partial U}{\partial y} - 2x \\ \dot{z} &= \frac{\partial U}{\partial z} \end{aligned} \tag{2}$$

$$r_1 = \sqrt{(x+\mu)^2 + y^2 + z^2}, \tag{3}$$

$$r_2 = \sqrt{(x-1+\mu)^2 + y^2 + z^2}. \tag{4}$$

All the Lagrangian points can then be found by posing:

$$\frac{\partial U}{\partial x} = \frac{\partial U}{\partial y} = \frac{\partial U}{\partial z} = 0 \tag{5}$$

First the collinear points L1, L2 and L3 are computed, which are found by searching particular solutions with $y = z = 0$. This condition satisfies $\frac{\partial U}{\partial y} = \frac{\partial U}{\partial z} = 0$. Therefore the problem of finding the collinear points positions can be rewritten as finding the three x coordinates that solve the following equation:

$$\left.\frac{\partial U}{\partial x}\right|_{y=0,z=0} = x - \frac{(1-\mu)(x+\mu)}{|x+\mu|^3} - \frac{\mu(x+\mu-1)}{|x-\mu+1|^3} = 0 \tag{6}$$

Using MATLAB's *fzero* function on the Eq. 6 expression and initial guesses chosen with a graphical approach on the plot of Eq. (6) (Figure 1), the x coordinates of the collinear points can be computed, while the y coordinates are always 0.
In order to find L4 and L5 lagrangian triangular points it can be observed that the conditions $r_1 = r_2 = 1$ and $z = 0$ also satisfy Eq. 5. This means graphically that the two points are located in the intersections between two unitary circles respectively centered on Earth and Moon (Figure 2). By reutilizing MATLAB's *fzero* function on the circles equations and initial guesses still based on a graphical approach the coordinates of L4 and L5 can be computed.
For the Jacobi constant calculation it has been used an auxiliary function that simply applies the formula.

	L_1	L_2	L_3	L_4	L_5
x	-1.0050624018	$+0.8369180073$	$+1.1556799131$	$+0.4878500000$	$+0.4878500000$
y	0.0000000000	0.0000000000	0.0000000000	0.8660254038	-0.8660254038
C	3.0241489429	3.2003380950	3.1841582164	3.0000000000	3.0000000000

Table 1: Lagrangian points coordinates and Jacobi constants

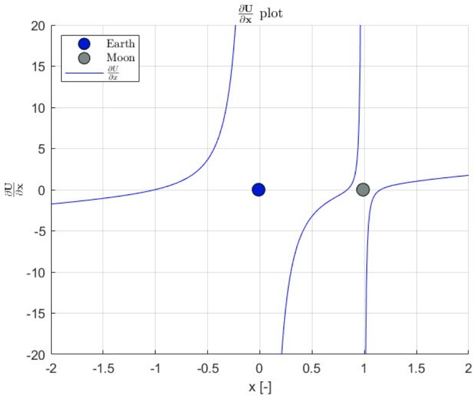

Figure 1: Plot for collinear points guesses in the rotating reference frame

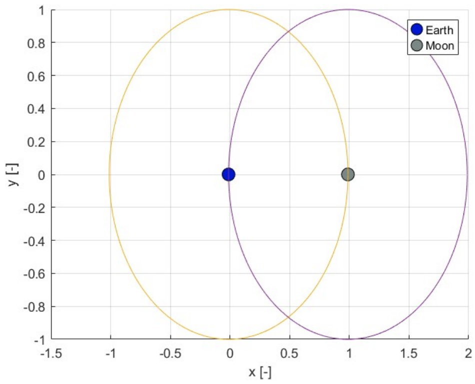

Figure 2: Plot for triangular points guesses in the rotating reference frame

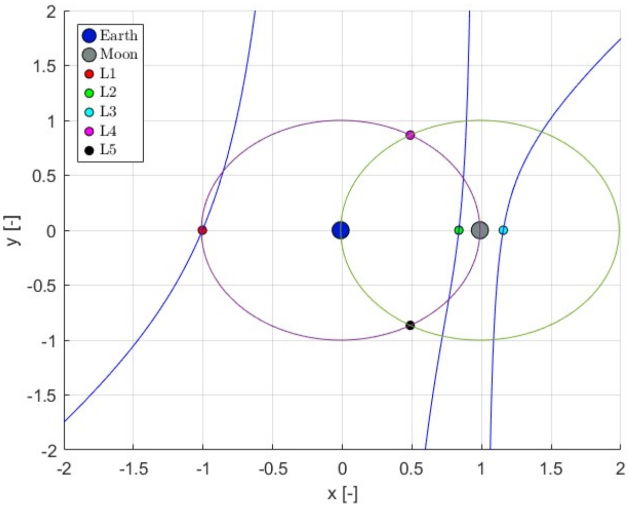

Figure 3: Earth-Moon lagrangian points location in the rotating reference frame

1.2 Point 2

To determine the correct initial conditions for a periodic halo orbit that has a Jacobi constant value of $C = 3.09$, a differential correction method was used. This approach adjusts the values of x_0, z_0 and v_{y0}, while keeping v_x and v_z set to zero at the point where the orbit crosses the $y = 0$ plane. This is necessary because, for the orbit to be periodic, it must cross the $y = 0$ plane orthogonally.

The motion is governed by the equations of the Circular Restricted Three Body Problem:

$$\begin{cases} \dot{x} = v_x \\ \dot{y} = v_y \\ \dot{z} = v_z \\ \dot{v}_x = 2v_y + x - \dfrac{(1-\mu)(x+\mu)}{[(x+\mu)^2 + y^2 + z^2]^{3/2}} - \dfrac{\mu(x+\mu-1)}{[(x+\mu-1)^2 + y^2 + z^2]^{3/2}} \\ \dot{v}_y = -2v_x + y - \dfrac{(1-\mu)y}{[(x+\mu)^2 + y^2 + z^2]^{3/2}} - \dfrac{\mu y}{[(x+\mu-1)^2 + y^2 + z^2]^{3/2}} \\ \dot{v}_z = -\dfrac{(1-\mu)z}{[(x+\mu)^2 + y^2 + z^2]^{3/2}} - \dfrac{\mu z}{[(x+\mu-1)^2 + y^2 + z^2]^{3/2}} \end{cases} \tag{7}$$

Starting from the chosen initial conditions, the orbit is integrated forward until it reaches the $y = 0$ plane. An event function is used to halt the integration precisely at this crossing.

The correction algorithm is based on Newton's method:

$$\mathbf{x_{k+1}} = \mathbf{x_k} - \boldsymbol{f}'(\mathbf{x_k})^{-1}\boldsymbol{f}(\mathbf{x_k}) \tag{8}$$

where $\mathbf{x}$ is the vector of unknowns, $\boldsymbol{f}(\mathbf{x})$ is the matrix of functions and $\boldsymbol{f}'(\mathbf{x})$ it's the Jacobian matrix. For this particular case the functions are the flow of the ODE at final time $\varphi(x_0, t_0, t_f)$ and C expression, while it has been considered for the Jacobian the variations with respect to the initial conditions and time, as suggested by the assignment:

$$\begin{pmatrix} \delta x_0 \\ \delta y_0 \\ \delta z_0 \\ \delta v_{x0} \\ \delta v_{y0} \\ \delta v_{z0} \\ \delta t \end{pmatrix} = \begin{bmatrix} \frac{\partial \varphi_x}{\partial x_0} & \cdots & \frac{\partial \varphi_x}{\partial v_{z0}} & \frac{\partial \varphi_x}{\partial t} \\ \vdots & \ddots & \vdots & \vdots \\ \frac{\partial \varphi_{v_z}}{\partial x_0} & \cdots & \frac{\partial \varphi_{v_z}}{\partial v_{z0}} & \frac{\partial \varphi_{v_z}}{\partial t} \\ \frac{\partial C}{\partial x_0} & \cdots & \frac{\partial C}{\partial v_{z0}} & \frac{\partial C}{\partial t} \end{bmatrix}^{-1} \begin{pmatrix} \delta x_f \\ \delta y_f \\ \delta z_f \\ \delta v_{xf} \\ \delta v_{yf} \\ \delta v_{zf} \\ \delta C \end{pmatrix} \tag{9}$$

In this expression there can be recognized the state transition matrix Φ as the upper left 6x6 submatrix inside the inverted Jacobian matrix, the right hand side of the CR3BP (7) on the first 6 elements of the last column and on the last row the partial derivatives of C expression. In order to compute the state transition matrix, the finite difference method has been used. To do so, the STM dynamics have been propagated solving the following system of ODE:

$$\begin{cases} \dot{\mathbf{\Phi}} & = \dfrac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}(t)\mathbf{\Phi} \\ \mathbf{\Phi}_0 & = \varphi(\boldsymbol{x}_0, t_0, t_0) = \mathbf{I}_{6\times 6} \end{cases} \tag{10}$$

Using a differential correction scheme the system of equations reads:

$$-\boldsymbol{J}(t_0, t_f) \begin{pmatrix} \Delta x_0 \\ 0 \\ \Delta z_0 \\ 0 \\ \Delta v_{y0} \\ 0 \\ \Delta t \end{pmatrix} = \begin{pmatrix} \tilde{x} \\ 0 \\ \tilde{z} \\ 0 \\ \tilde{v}_y \\ 0 \\ C_f \end{pmatrix} - \begin{pmatrix} x_f \\ y_f \\ z_f \\ v_{xf} \\ v_{yf} \\ v_{zf} \\ 3.09 \end{pmatrix} \tag{11}$$

Where $\tilde{\boldsymbol{x}}$ indicates the final state at the crossing, needed to achieve the Halo orbit with the desired C, and $\boldsymbol{J}$ it's the Jacobian matrix from eq. (9). Notice how the Jacobi constant correction it's opposite in sign, that's because the gradient of C it's moving toward increasing direction. So in order to have positive residuals make a correction the sign needs to be flipped. By extracting the useful components to solve the system, the problem reduces to:

$$\begin{pmatrix} \Delta x_0 \\ \Delta z_0 \\ \Delta v_{y0} \\ \Delta t \end{pmatrix} = \begin{bmatrix} \Phi_{21} & \Phi_{23} & \Phi_{25} & \frac{\partial \varphi_y}{\partial t} \\ \Phi_{41} & \Phi_{43} & \Phi_{45} & \frac{\partial \varphi_{v_x}}{\partial t} \\ \Phi_{61} & \Phi_{63} & \Phi_{65} & \frac{\partial \varphi_{v_z}}{\partial t} \\ \frac{\partial C}{\partial x_0} & \frac{\partial C}{\partial z_0} & \frac{\partial C}{\partial v_{y0}} & 0 \end{bmatrix}^{-1} \begin{pmatrix} y_f \\ v_{x_f} \\ v_{z_f} \\ 3.09 - \mathrm{C}_f \end{pmatrix}$$

The correction on the encounter time it's not used in the next step, since it was set a reasonably high final time of integration in the numerical solver, in order to surely encounter x-axis. The final expression used to compute the correct values of the initial state is:

$$\begin{pmatrix} x_0 \\ z_0 \\ v_{y0} \end{pmatrix}_{k+1} = \begin{pmatrix} x_0 \\ z_0 \\ v_{y0} \end{pmatrix}_k - \begin{pmatrix} \Delta x_0 \\ \Delta z_0 \\ \Delta v_{y0} \end{pmatrix}$$

With an error tolerance of 10^{-10} the results are:

x	y	z
+1.0590402078	0.0000000000	+0.0739277378

v_x	v_y	v_z
0.0000000000	+0.3469245707	0.0000000000

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

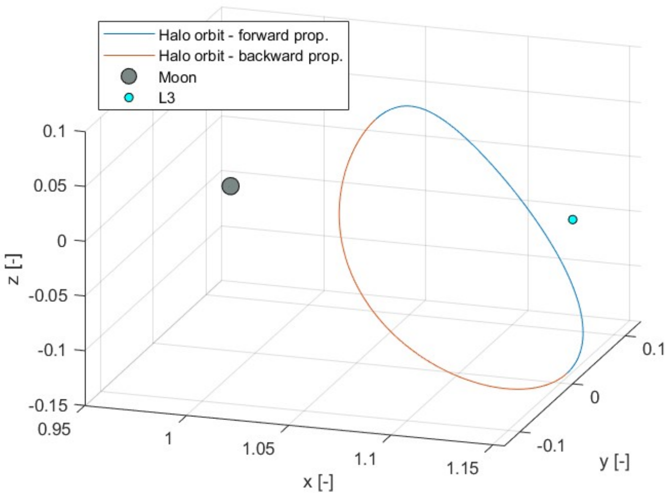

Figure 4: $C = 3.09$ Halo orbit plot in the rotating reference frame

1.3 Point 3

To ensure fast convergence of the algorithm, numerical continuation was used. Specifically, a series of five orbits, each with a Jacobi constant C spaced evenly between 3.09 and 3.04, was considered. Starting with the initial guess from point 2, the algorithm found the correct initial state for the first orbit. This result then served as the starting point for the next orbit in the series, and the process was repeated. In this way, the correct initial state for a periodic halo orbit with C = 3.04 was finally obtained.

x	y	z
+1.0125655235	0.0000000000	+0.0672339583

v_x	v_y	v_z
0.0000000000	+0.5103251956	0.0000000000

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

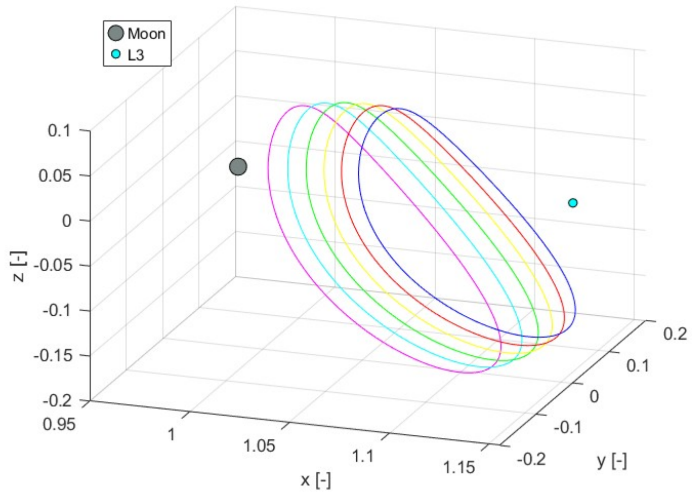

Figure 5: Halo orbits familiy from $C = 3.09$ to $C = 3.04$ in the rotating reference frame

2 Impulsive guidance

Exercise 2

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

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

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

Table 4: Constants to be considered to solve the PBRFBP. The units of distance, time, and velocity are used to map scaled quantities into physical units.

2) Considering the first guess in 1) and using $\{\mathbf{x}_i, t_i, t_f\}$ as variables, solve the problem in Section 3.1 with simple shooting in the following cases

    a) without providing any derivative to the solver, and

*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.

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.
   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 Point 1

Given $\{\alpha, \beta, t_i, \delta\}$, the construction of the initial guess of the solution showed in Section 3.2 * requires to impose $r_0 = r_i$ and $v_0 = \beta\sqrt{(1-\mu)/r_0}$, so that the initial transfer state, $x_0(\alpha, \beta) = (x_0, y_0, \dot{x}_0, \dot{y}_0)$, reads:

$$x_0 = r_0 \cos\alpha - \mu, \quad y_0 = r_0 \sin\alpha, \quad \dot{x}_0 = -(v_0 - r_0)\sin\alpha, \quad \dot{y}_0 = (v_0 - r_0)\cos\alpha \tag{12}$$

The following table shows the resulting initial guess with the assigned data:

$r_{x,0}$ $[DU]$	$r_{y,0}$ $[DU]$	$v_{x,0}$ $[VU]$	$v_{y,0}$ $[VU]$
+0.0016242	+0.0100080	-6.3027381	+8.6749748

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

An orbit in the rotating frame, $\mathbf{x}(t) = \{x(t), y(t), \dot{x}(t), \dot{y}(t)\}$, is converted to an orbit expressed in the P_1-centered (or Earth-centered), inertial frame, $\mathbf{X}_1(t) = \{X_1(t), Y_1(t), \dot{X}_1(t), \dot{Y}_1(t)\}$, through

$$\begin{aligned}
X_1(t) &= (x(t) + \mu)\cos t - y(t)\sin t \\
Y_1(t) &= (x(t) + \mu)\sin t + y(t)\cos t \\
\dot{X}_1(t) &= (\dot{x}(t) - y(t))\cos t - (\dot{y}(t) + x(t) + \mu)\sin t \\
\dot{Y}_1(t) &= (\dot{x}(t) - y(t))\sin t + (\dot{y}(t) + x(t) + \mu)\cos t
\end{aligned} \tag{13}$$

where t is the scaled instant of conversion. To assess that the conversion was done correctly the difference in norm between the position vectors in ECI and rotating frame was evaluated for each instant, showing good low error values.

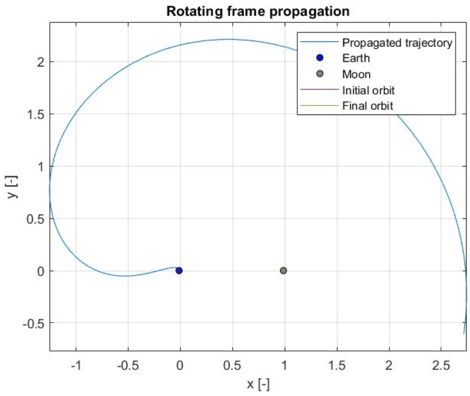

Figure 6: Propagation of the guess in the rotating frame

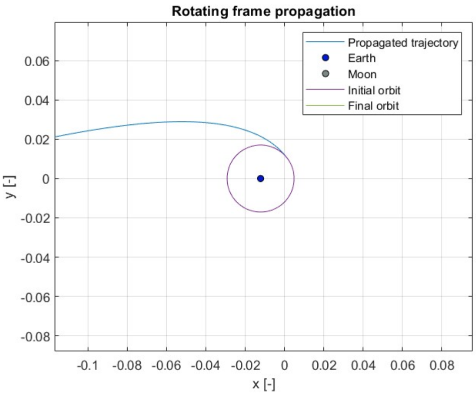

Figure 7: Detail on Earth's starting orbit in rotating frame

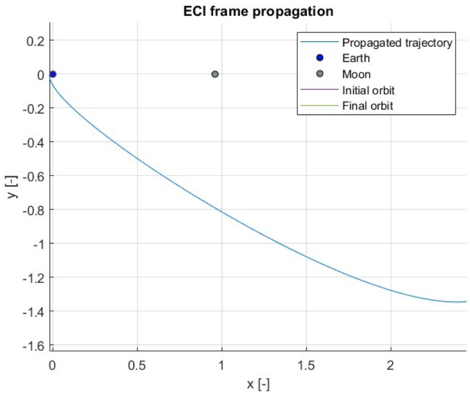

Figure 8: Propagation of the guess in the ECI frame

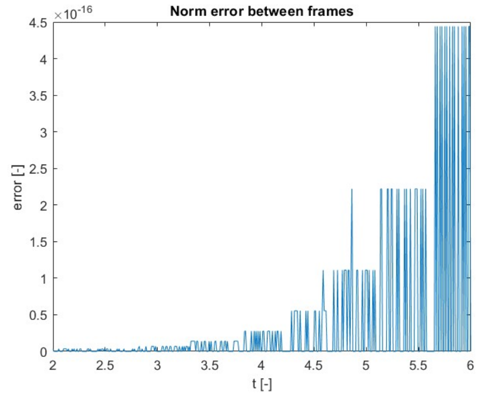

Figure 9: From rotating frame to ECI conversion error trend vs time

2.2 Point 2

The objective of this point is to find the initial state of the spacecraft that minimizes the Δv of an Earth-Moon transfer, through an impulsive maneuver and a simple shooting algorithm. The dynamics of the problem it's described by the PBRFBP equations:

$$\ddot{x} - 2\dot{y} = \frac{\partial \Omega_4}{\partial x}, \qquad \ddot{y} + 2\dot{x} = \frac{\partial \Omega_4}{\partial y}, \tag{14}$$

where the four-body effective potential, Ω_4, reads:

$$\Omega_4(x, y, t) = \Omega_3(x, y) + \frac{m_s}{r_3(t)} - \frac{m_s}{\rho^2}(x\cos(\omega_s t) + y\sin(\omega_s t)). \tag{15}$$

And the solution integrated from this equations is $\mathbf{x}(t) = \varphi(\mathbf{x}_i, t_i; t)$, with $t > t_i$.
The problem statement is formalized through typical Non-Linear programming principles:

$$\min_{\mathbf{y}} J \quad \text{s.t.} \quad \begin{cases} \mathbf{h}(\mathbf{y}) \leq 0 \\ \mathbf{g}(\mathbf{y}) = 0 \end{cases} \tag{16}$$

where $\mathbf{y}$ is the set of optimization variables:

$$\mathbf{y} = (\mathbf{x}_i,\, t_i,\, t_f) \tag{17}$$

with $\mathbf{x}_i$ being the state of the spacecraft at the departure time t_i, and t_f the arrival time.
The objective function to minimize describes the sum of Δv expended to leave the starting Earth orbit and park to the arrival Moon orbit:

$$J = \Delta v(\mathbf{x}_i,\, t_i,\, t_f) = \Delta v_i(\mathbf{x}_i) + \Delta v_f\left(\varphi(\mathbf{x}_i,\, t_i;\, t_f)\right) \tag{18}$$

The Δv are computed as:

$$\Delta v_i = \sqrt{(\dot{x}_i - \dot{y}_i)^2 + (\dot{y}_i + x_i + \mu)^2} - \sqrt{\frac{1-\mu}{r_i}}, \tag{19}$$

$$\Delta v_f = \sqrt{(\dot{x}_f - \dot{y}_f)^2 + (\dot{y}_f + x_f + \mu - 1)^2} - \sqrt{\frac{\mu}{r_f}}, \tag{20}$$

The equality constraints contained in vector $\mathbf{g}(\mathbf{y})$ are on the initial and arrival state:

$$\boldsymbol{\psi}_i := \begin{bmatrix} (x_i+\mu)^2 + y_i^2 - r_i^2 \\ (x_i+\mu)(\dot{x}_i - \dot{y}_i) + y_i(\dot{y}_i + x_i + \mu) \end{bmatrix} = 0 \tag{21}$$

$$\boldsymbol{\psi}_f := \begin{bmatrix} (x_f+\mu-1)^2 + y_f^2 - r_f^2 \\ (x_f+\mu-1)(\dot{x}_f - \dot{y}_f) + y_f(\dot{y}_f + x_f + \mu - 1) \end{bmatrix} = 0 \tag{22}$$

while the inequality constraint contained in $\mathbf{h}(\mathbf{y})$ is on the times:

$$t_i - t_f < 0 \tag{23}$$

To solve this problem it was implemented a MATLAB algorithm that minimizes J through *fmincon*. The gradients provided to the solver are:

$$\nabla J = \begin{pmatrix} \frac{\partial J}{\partial \mathbf{x_i}} \\ \frac{\partial J}{\partial t_i} \\ \frac{\partial J}{\partial t_f} \end{pmatrix} = \begin{pmatrix} -\mathbf{\Phi}_{\mathrm{vr}}^T \frac{\Delta \mathbf{v}_f}{\|\Delta \mathbf{v}_f\|} \\ \frac{\Delta \mathbf{v}_i}{\|\Delta \mathbf{v}_i\|} - \mathbf{\Phi}_{\mathrm{vv}}^T \frac{\Delta \mathbf{v}_f}{\|\Delta \mathbf{v}_f\|} \\ \frac{\Delta \mathbf{v}_i}{\|\Delta \mathbf{v}_i\|}\left(-\dot{\mathbf{v}}_f(t_f)\right) + \frac{\Delta \mathbf{v}_f}{\|\Delta \mathbf{v}_f\|}\left(\mathbf{\Phi}_{\mathrm{vr}}\mathbf{f}_r + \mathbf{\Phi}_{\mathrm{vv}}\mathbf{f}_v\right) \\ \frac{\Delta \mathbf{v}_f}{\|\Delta \mathbf{v}_f\|}\left(\dot{\mathbf{v}}(t_f) - \mathbf{f}_v(\mathbf{x}(t_f), t_f)\right) \end{pmatrix}$$

with $\mathbf{\Phi}$ being the STM matrix:

$$\mathbf{\Phi} = \begin{pmatrix} \mathbf{\Phi}_{\mathrm{rr}} & \mathbf{\Phi}_{\mathrm{rv}} \\ \mathbf{\Phi}_{\mathrm{vr}} & \mathbf{\Phi}_{\mathrm{vv}} \end{pmatrix} \tag{24}$$

and $\mathbf{f}_r$ and $\mathbf{f}_v$ being the right hand sides of the orbital dynamic described in Eq (14), respectively position and velocity.
The results are:

Gradients	$r_{x,0}$ $[DU]$	$r_{y,0}$ $[DU]$	$v_{x,0}$ $[VU]$	$v_{y,0}$ $[VU]$	t_i $[TU]$	t_f $[TU]$	Δv $[VU]$
False	+0.001635	+0.009993	-6.277868	+8.660350	2.323	6.322	4.00768
True	+0.001620	+0.010013	-6.290740	+8.651414	2.198	6.203	4.00683

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

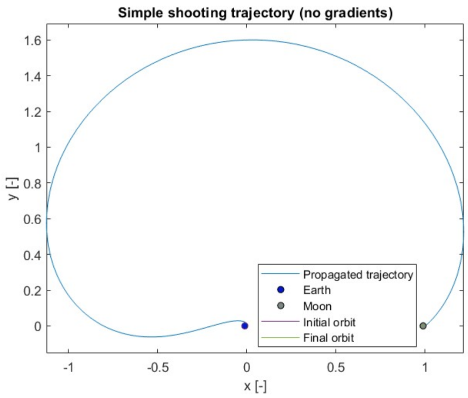

Figure 10: Propagation of the simple shooting solution with no gradients provided in the rotating reference frame

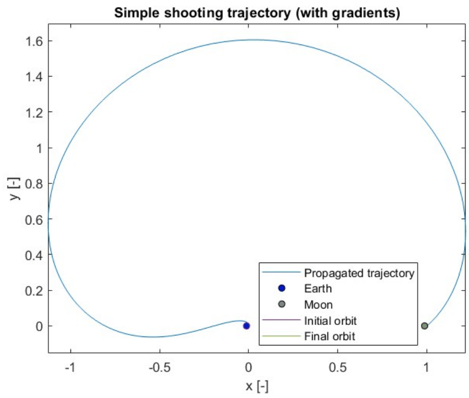

Figure 11: Propagation of the simple shooting solution with gradients provided in the rotating reference frame

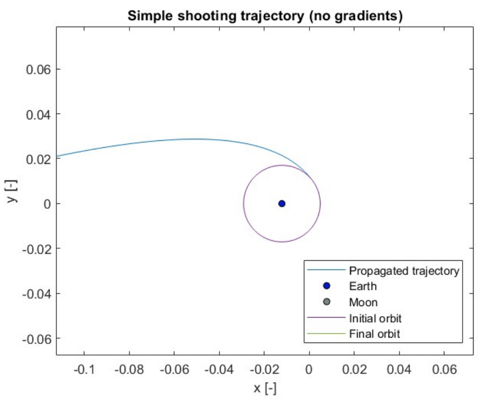

Figure 12: Initial orbit detail of no gradients trajectory in the rotating reference frame

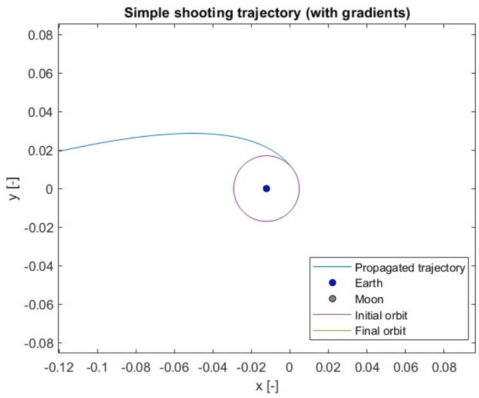

Figure 13: Initial orbit detail of gradients trajectory in the rotating reference frame

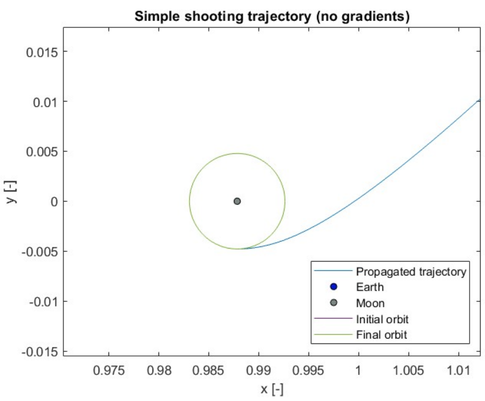

Figure 14: Final orbit detail of no gradients trajectory in the rotating reference frame

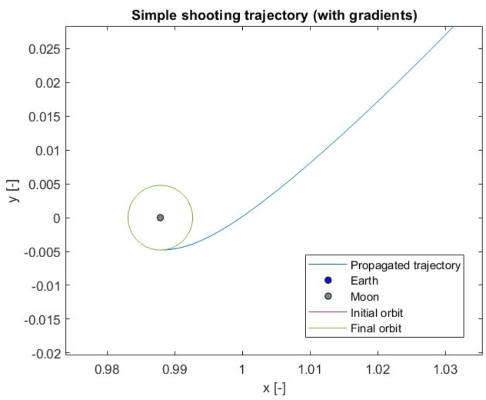

Figure 15: Final orbit detail of gradients trajectory in the rotating reference frame

2.3 Point 3

To solve this point a multiple shooting algorithm was implemented in MATLAB. Let N be the number of segments, so that in this case the optimization variables are:

$$\mathbf{y} = (\mathbf{x}_i, .., \mathbf{x}_{N-1}, \mathbf{x}_N, t_i,\, t_f) \tag{25}$$

with $\mathbf{x}_N = \mathbf{x}_f$ and the states between $\mathbf{x}_i$ and $\mathbf{x}_N$ are the intermediate shooting states. The objective function J remains the same of the previous point, while it's gradient it's now expressed like:

$$\frac{\partial J}{\partial \mathbf{y}} = \begin{bmatrix} \boldsymbol{P}_1 & \boldsymbol{O} & \boldsymbol{P}_N & \boldsymbol{O} \end{bmatrix}, \tag{26}$$

where

$$\boldsymbol{P}_1 := \frac{\partial J}{\partial \boldsymbol{x}_1} = \frac{1}{\sqrt{(\dot{x}_1 - \dot{y}_1)^2 + (\dot{y}_1 + x_1 + \mu)^2}} \begin{bmatrix} \dot{y}_1 + x_1 + \mu, & y_1 - x_1, & x_1 - y_1, & y_1 + x_1 + \mu \end{bmatrix}, \tag{27}$$

$$\boldsymbol{P}_N := \frac{\partial J}{\partial \boldsymbol{x}_N} = \frac{1}{\sqrt{(\dot{x}_N - \dot{y}_N)^2 + (\dot{y}_N + x_N + \mu - 1)^2}} \begin{bmatrix} \dot{y}_N + x_N + \mu - 1, & y_N - x_N, & x_N - y_N, & \dot{y}_N + x_N + \mu - 1 \end{bmatrix} \tag{28}$$

This gradient, and the other that will follow in this report, are based on Appendix 2 (Oshima, Topputo, Yamao, 2019)[†].
The equality constraints of the problem are on the initial state, the final state and the continuity of the trajectory between intermediate states:

$$\boldsymbol{\psi}_i := \begin{bmatrix} (x_i + \mu)^2 + y_i^2 - r_i^2 \\ (x_i + \mu)(\dot{x}_i - \dot{y}_i) + y_i(\dot{y}_i + x_i + \mu) \end{bmatrix} = 0 \tag{29}$$

$$\boldsymbol{\psi}_f := \begin{bmatrix} (x_f + \mu - 1)^2 + y_f^2 - r_f^2 \\ (x_f + \mu - 1)(\dot{x}_f - \dot{y}_f) + y_f(\dot{y}_f + x_f + \mu - 1) \end{bmatrix} = 0 \tag{30}$$

$$\boldsymbol{\zeta}_j := \varphi(\mathbf{x_j}, t_j, t_{j+1}) - \mathbf{x}_{j+1} = 0, \quad j = 1, \dots, N-1 \tag{31}$$

The derivative of the equality constraints $\boldsymbol{c} := \{\boldsymbol{\zeta}_j, \boldsymbol{\psi}_1, \boldsymbol{\psi}_N\} = 0$ with respect to the NLP variables $\mathbf{y}$ it's computed as:

$$\frac{\partial \boldsymbol{c}}{\partial \mathbf{y}} = \begin{bmatrix} \boldsymbol{\Phi}(t_1, t_2) & -\mathbf{I}_4 & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \mathbf{Q}_1^1 & \mathbf{Q}_N^1 \\ \boldsymbol{O} & \boldsymbol{\Phi}(t_2, t_3) & -\mathbf{I}_4 & \boldsymbol{O} & \boldsymbol{O} & \mathbf{Q}_1^2 & \mathbf{Q}_N^2 \\ \boldsymbol{O} & \boldsymbol{O} & \ddots & \ddots & \ddots & \vdots & \vdots \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{\Phi}(t_{N-1}, t_N) & -\mathbf{I}_4 & \mathbf{Q}_1^{N-1} & \mathbf{Q}_N^N - 1 \\ \mathbf{R}_1 & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \mathbf{R}_N & \boldsymbol{O} & \boldsymbol{O} \end{bmatrix}, \tag{32}$$

where:

$$\mathbf{Q}_1^j := \frac{\partial \boldsymbol{\zeta}_j}{\partial t_1} = -\frac{N-j}{N-1}\boldsymbol{\Phi}(t_j, t_{j+1})\mathbf{f}(\boldsymbol{x_j}, t_j) + \frac{N-j-1}{N-1}\mathbf{f}(\varphi(\mathbf{x}_j, t_j, t_{j+1}), t_{j+1}), \quad j = 1, \dots, N-1, \tag{33}$$

$$\mathbf{Q}_N^j := \frac{\partial \boldsymbol{\zeta}_j}{\partial t_N} = -\frac{j-1}{N-1}\boldsymbol{\Phi}(t_j, t_{j+1})\mathbf{f}(\mathbf{x}_j, t_j) + \frac{j}{N-1}\mathbf{f}(\varphi(\mathbf{x}_j, t_j, t_{j+1}), t_{j+1}), \quad j = 1, \dots, N-1, \tag{34}$$

[†]K. Oshima, F. Topputo, T. Yanao, "Low-Energy Transfers to the Moon with Long Transfer Time ", Celestial Mechanics and Dynamical Astronomy, Vol. 131, pp. 1–19, 2019, DOI: 10.1007/s10569-019-9883-7 .

$$\mathbf{R}_1 := \frac{\partial \boldsymbol{\psi}_1}{\partial \mathbf{x_1}} = \begin{bmatrix} 2(x_1+\mu) & 2y_1 & 0 & 0 \\ \dot{x}_1 & y_1 & x_1+\mu & y_1 \end{bmatrix} \tag{35}$$

$$\mathbf{R}_N := \frac{\partial \boldsymbol{\psi}_N}{\partial \mathbf{x_N}} = \begin{bmatrix} 2(x_N+\mu-1) & 2y_N & 0 & 0 \\ \dot{x}_N & y_N & x_N+\mu-1 & y_N \end{bmatrix} \tag{36}$$

The results are:

$r_{x,0}$ $[DU]$	$r_{y,0}$ $[DU]$	$v_{x,0}$ $[VU]$	$v_{y,0}$ $[VU]$	t_i $[TU]$	t_f $[TU]$	Δv $[VU]$
+0.004070	+0.005176	-3.251054	+10.188317	2.462	6.127	3.997338

Table 7: Multiple shooting solution in the Earth-Moon rotating frame.

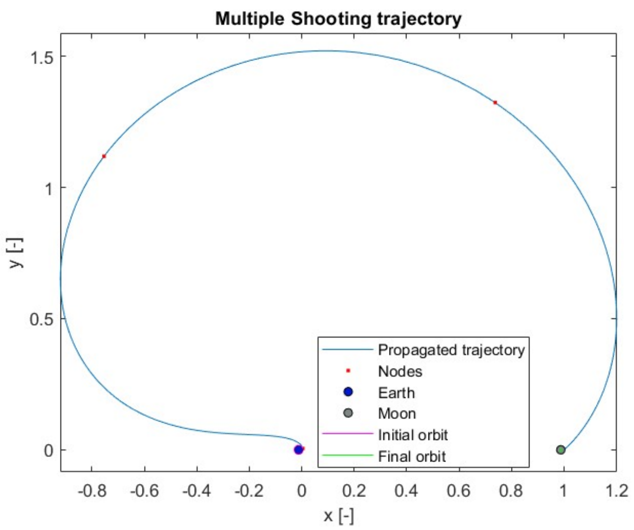

Figure 16: Propagation of the multiple shooting solution in the rotating reference frame

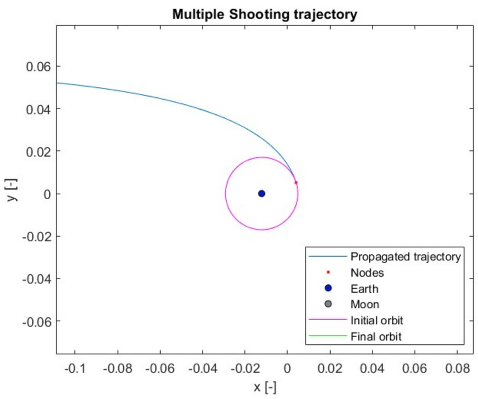

Figure 17: Initial orbit detail of multiple shooting trajectory in the rotating reference frame

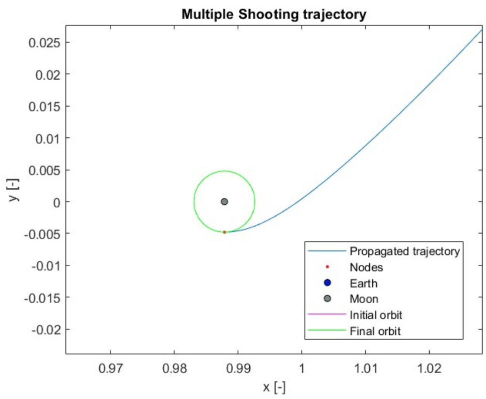

Figure 18: Final orbit detail of multiple shooting trajectory in the rotating reference frame

2.4 Point 4

The angle at the departing time of the spacecraft it's defined as:

$$\theta_i = \omega_s t_i \tag{37}$$

This value is used to create a MATLAB algorithm that studies the roots of the function:

$$\Delta\theta(t) = \theta(t) - \theta_i \tag{38}$$

where $\theta(t)$ is the angle between the position vector $\boldsymbol{r}_{EMB-Moon}$ and the projection of the position vector $\boldsymbol{r}_{EMB-Sun}$ on the plane containing the EMB and the Moon, and in order to consider it in the $[0,2\pi]$ range it is computed it this convention:

$$\theta(t) = \begin{cases} acos(\frac{\boldsymbol{r}_{EMB-Moon} \cdot \boldsymbol{r}_{EMB-Sun}}{r_{EMB-Moon} r_{EMB-Sun}}), & \boldsymbol{r}_{EMB-Moon} \times \boldsymbol{r}_{EMB-Sun} \geq 0 \\ 2\pi - acos(\frac{\boldsymbol{r}_{EMB-Moon} \cdot \boldsymbol{r}_{EMB-Sun}}{r_{EMB-Moon} r_{EMB-Sun}}), & \boldsymbol{r}_{EMB-Moon} \times \boldsymbol{r}_{EMB-Sun} < 0 \end{cases} \tag{39}$$

All the positions in time are retrieved through the ephemerids provided by SPICE's function *cspice_spkezr*. As we can see from Figure 19 this function crosses multiple times the x-axis, in this case in a 4 month long timespan. In order to find the correct date it has been used the function $fzero$ and set a proper dates range (Figure 20) that analizes a monotonical trait of $\Delta\theta$. The dates range was chosen near the suggested guess of the 28/9/2024, and it was verified with a graphical approach if the zero-found solution was actually leading near the moon trajectory at the final time. However due to the approximation of using the angle between the Moon position and a projection of the Sun position, and due to the fact that equations in 13 are very sensitive to the present scaled time inputed the tarting obtained it's not the required one, and the n-body yields trajectory that doesn't encounter the Moon at the end (Figure 21). Results are:

Symbol	**Calendar epoch** (UTC)
t_i	2024-10-12T21:18:34.082
t_f	2024-10-30T06:43:20.260

$r_{x,0}$ $[km]$	$r_{y,0}$ $[km]$	$r_{z,0}$ $[km]$
-6223.6241	$+2026.1578$	0.0

$v_{x,0}$ $[km/s]$	$v_{y,0}$ $[km/s]$	$v_{z,0}$ $[km/s]$
-3.3980	-10.4374	0.0

Table 8: Initial epoch, final epoch, and initial state in Earth-centered inertial frame.

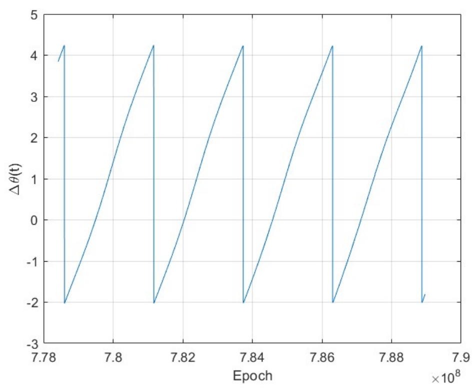

Figure 19: $\Delta\theta(t)$ from 1/9/2024 to 1/1/2025 (plot with discontinuities)

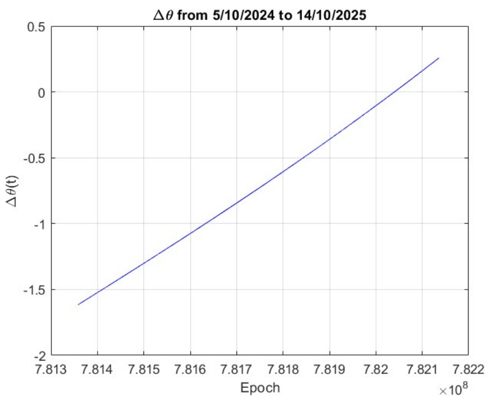

Figure 20: $\Delta\theta(t)$ in the restricted time interval

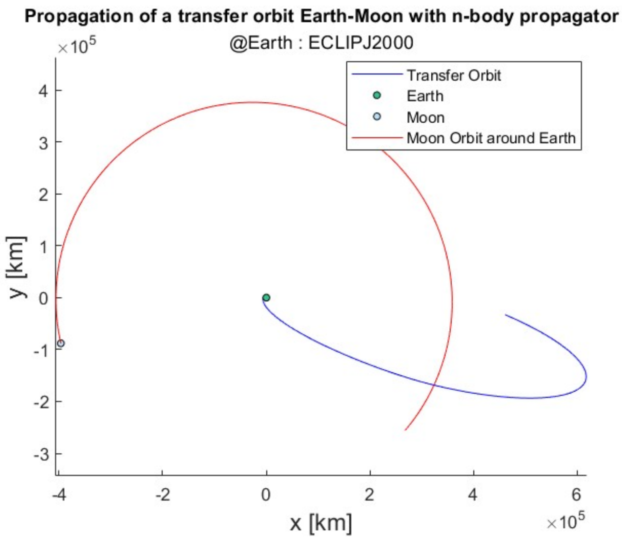

Figure 21: Graphical verification of the Moon encounter with the zero-finding solution

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 9.

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 9: 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 9. 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 Point 1

The initial and final state are computed as:

$$\boldsymbol{x}_i = \begin{bmatrix} r_E + h_i \\ 0 \\ 0 \\ 0 \\ \sqrt{\frac{\mu}{R_e+h_i}} \\ 0 \end{bmatrix}, \quad \boldsymbol{x}_f = \begin{bmatrix} r_E + h_f \\ 0 \\ 0 \\ 0 \\ \sqrt{\frac{\mu}{R_e+h_i}} cos(\Delta i) \\ \sqrt{\frac{\mu}{R_e+h_i}} sin(\Delta i) \end{bmatrix} \tag{40}$$

With this computation the initial state and final lie on the x axis of the equatorial J2000 reference frame, and to the final velocity it's applied the rotation angle Δi. The results are:

$r_{x,i}$ $[km]$	$r_{y,i}$ $[km]$	$r_{z,i}$ $[km]$	$v_{x,i}$ $[km/s]$	$v_{y,i}$ $[km/s]$	$v_{z,i}$ $[km/s]$
+7178.1366	0	0	0	+7.4518314	0

$r_{x,f}$ $[km]$	$r_{y,f}$ $[km]$	$r_{z,f}$ $[km]$	$v_{x,f}$ $[km/s]$	$v_{y,f}$ $[km/s]$	$v_{z,f}$ $[km/s]$
+7378.1366	0	0	0	+7.3495090	+0.0962103

Table 10: Initial and target state in Earth-centered equatorial J2000 intertial frame.

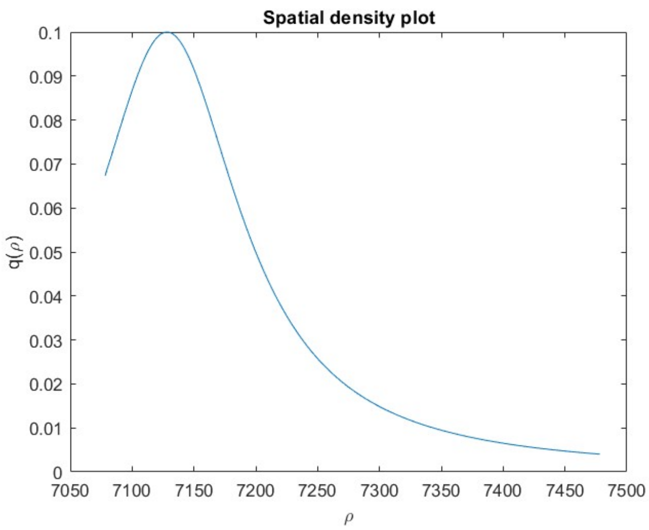

Figure 22: Spatial density plot vs distance from Earth

[‡]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.

3.2 Point 2

The gravitational parameter μ is measured in km^3/s^2. By imposing $\mu_{ad} = 1$ the adimensionalizing time unit TU is computed as:

$$\mu_{ad} = \mu\frac{TU^2}{LU^3} \rightarrow TU = \sqrt{\frac{LU^3}{\mu}} \tag{41}$$

The consequent velocity unit is:

$$VU = \frac{DU}{TU} \tag{42}$$

and the adimensionalized parameters are computed as:

$$\begin{aligned}
r_{i,ad} &= \frac{Re + h_i}{DU}, & r_{f,ad} &= \frac{Re + h_f}{DU},\\
\rho_{0,ad} &= \frac{\rho_0}{DU}, & T_{max,ad} &= \frac{T_{max} \cdot TU^2}{MU \cdot DU \cdot 1000},\\
I_{s,ad} &= \frac{I_s}{TU}, & v_{i,ad} &= \frac{v_i}{VU},\\
v_{f,ad} &= \frac{v_f}{VU}, & g_{0,ad} &= \frac{g_0 \cdot TU^2}{1000 \cdot DU}
\end{aligned}$$

3.3 Point 3

In order to apply the Pontryagin Maximum Principle it's necessary to define the Hamiltonian function of the problem:

$$H = l + \boldsymbol{\lambda}^T \boldsymbol{f} \tag{43}$$

with l being the objective function to minimize, $\boldsymbol{f}$ the problem dynamics described in $\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}, t)$, and $\boldsymbol{\lambda}$ the vector of costates. In this case:

$$\boldsymbol{f} = \begin{cases} \dot{\mathbf{r}} &= \mathbf{v}\\ \dot{\mathbf{v}} &= -\frac{\mu}{r^3}\mathbf{r} + \mu\frac{T_{\max}}{m}\hat{\boldsymbol{\alpha}}\\ \dot{m} &= -\mu\frac{T_{\max}}{I_{\mathrm{sp}}g_0}\end{cases} \quad \boldsymbol{\lambda} = \begin{bmatrix}\boldsymbol{\lambda_r}\\ \boldsymbol{\lambda_v}\\ \lambda_m\end{bmatrix} \quad l = q(\rho) \tag{44}$$

The resulting spacecraft equation of motion are obtained as:

$$\begin{cases}\dot{\boldsymbol{x}} = \frac{\partial H}{\partial \boldsymbol{\lambda}}\\ \dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \boldsymbol{x}}\end{cases} \rightarrow \begin{cases} \dot{\mathbf{r}} &= \mathbf{v}\\ \dot{\mathbf{v}} &= -\frac{\mu}{r^3}\mathbf{r} + u\frac{T_{\max}}{m}\hat{\boldsymbol{\alpha}}\\ \dot{m} &= -u\frac{T_{\max}}{I_{\mathrm{sp}}g_0}\\ \dot{\boldsymbol{\lambda}}_r &= -\frac{3\mu}{r^5}(\mathbf{r}\cdot\boldsymbol{\lambda}_v)\mathbf{r} + \frac{\mu}{r^3}\boldsymbol{\lambda}_v + 2k_1\frac{(\mathbf{r}-\boldsymbol{\rho}_0)}{((\mathbf{r}-\boldsymbol{\rho}_0)^2+k_2)^2}\mathbf{r}\\ \dot{\boldsymbol{\lambda}}_v &= -\boldsymbol{\lambda}_r\\ \dot{\lambda}_m &= -u\frac{T_{\max}}{I_{\mathrm{sp}}g_0}\boldsymbol{\lambda}_v\cdot\hat{\boldsymbol{\alpha}}\end{cases} \tag{45}$$

where $\hat{\boldsymbol{\alpha}} = -\frac{\boldsymbol{\lambda}_v}{\|\boldsymbol{\lambda}_v\|}$ and $u = 1$, since it's the control input value that always keeps the Hamiltonian minimized . The boundary conditions and the transversality condition are respectively:

$$\begin{cases}\boldsymbol{r}(t_f) = \boldsymbol{r}_f\\ \boldsymbol{v}(t_f) = \boldsymbol{v}_f\\ \lambda_m = 0\end{cases} \quad \text{and} \quad H(t_f) = \frac{k_1}{k_2 + (r-\rho_0)^2} + \boldsymbol{\lambda}(t_f)\cdot\dot{\mathbf{x}} = 0 \tag{46}$$

3.4 Point 4

To solve this problem numerically an algorithm that exploits MATLAB's $fsolve$ function was implemented. This algorithm takes inside a *while* cycle the randomly generated guesses in the suggested interval as input for $fsolve$, where the guesses of $\boldsymbol{\lambda}_i$ and t_f are propagated to the final time. The final state obtained and the Hamiltonian at the final time are then subtracted to the boundary conditions and the transversality condition. In order to find the correct solution the difference has to be null. The results are:

t_f [mins]	m_f $[kg]$
1035.1973	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 11: Optimal orbit raising transfer solution ($T_{\max} = 3.000$ N).

Error	**Value**	**Units**
$\|\|\mathbf{r}(t_f) - \mathbf{r}_f\|\|$	0.5104	km
$\|\|\mathbf{v}(t_f) - \mathbf{v}_f\|\|$	0.0000005093	m/s

Table 12: Final state error with respect to target position and velocity ($T_{\max} = 3.000$ N).

This results were validated by checking that the value of the hamiltonian function stays null for the whole time of the transfer, as the final value has to be 0 and H in this case is constant. From Figure 26 it can be seen that the Hamiltonian is constant around zero.

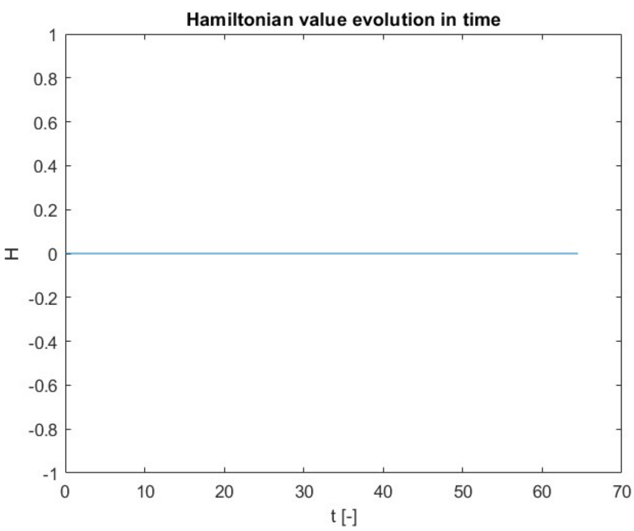

Figure 23: 3N solution H evolution in time

Below are the plots of $\hat{\boldsymbol{\alpha}}$ components:

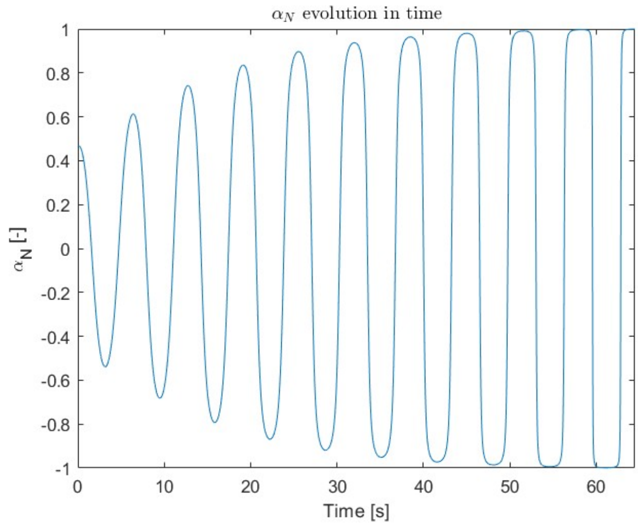

Figure 24: α_N component evolution in time

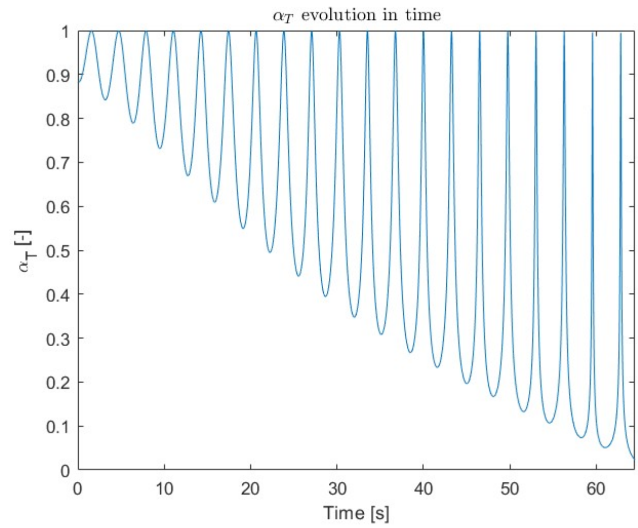

Figure 25: α_T component evolution in time

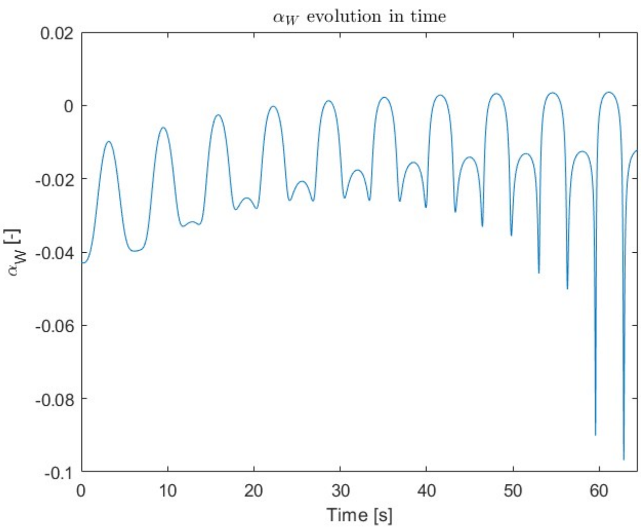

Figure 26: α_W component evolution in time

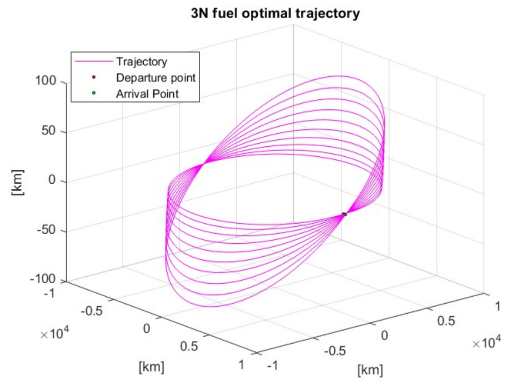

Figure 27: 3N fuel optimal trajectory in the ECI frame

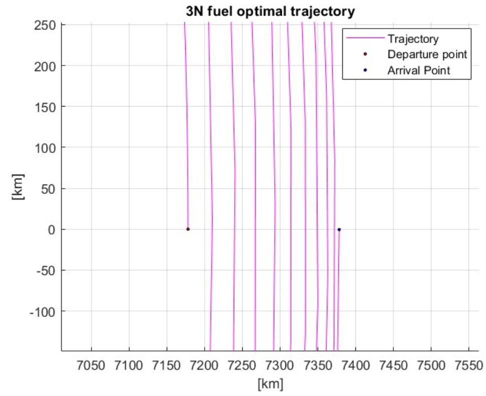

Figure 28: 3N fuel optimal trajectory zoom on departure and arrival point on xy plane of ECI frame

3.5 Point 5

The was also solved for the 2.860 N case, with the exact same procedure of the previous section. To have a faster convergence of the solver, numerical continuation was utilized. An equally spaced thrust level vector, consisting of 100 elements ranging from $3N$ to $2.860N$, was created. The solution for each thrust level was found by using the previous component's solution as the initial guess of $fsolve$.

t_f [mins]	m_f $[kg]$
1031.5144	994.2167

λ_{0,r_x}	λ_{0,r_y}	λ_{0,r_z}	λ_{0,v_x}	λ_{0,v_y}	λ_{0,v_z}	$\lambda_{0,m}$
-425.9501	-11.2531	$+1.2846$	-11.4431	-425.5786	-547.3406	$+10.4493$

Table 13: Optimal orbit raising transfer solution ($T_{\max} = 2.860$ N).

Error	**Value**	**Units**
$\|\|\mathbf{r}(t_f) - \mathbf{r}_f\|\|$	0.3130	km
$\|\|\mathbf{v}(t_f) - \mathbf{v}_f\|\|$	0.0000004049	m/s

Table 14: Final state error with respect to target position and velocity ($T_{\max} = 2.860$ N).

From the results it can be observed that with this value of maximum thrust there's less consumption on fuel and the error on the final state is lower. The results were assessed by plotting the evolution of H (Figure 29). It can be seen that the Hamiltonian function still stays around 0, with a lower precision because of the numerical issues caused by the numerical continuation method.

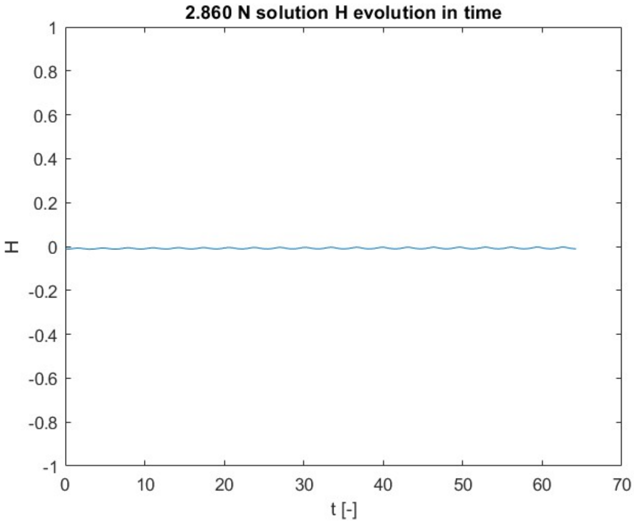

Figure 29: 2.860 H evolution in time

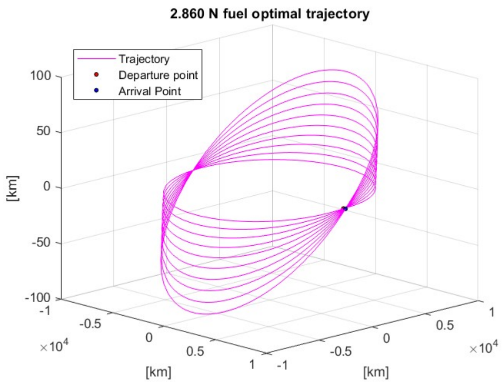

Figure 30: 2.860N fuel optimal trajectory in the ECI frame

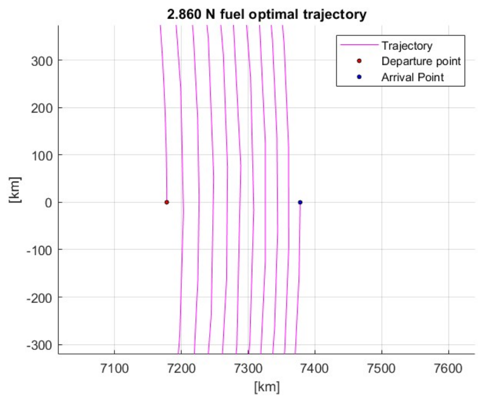

Figure 31: 3N fuel optimal trajectory zoom on departure and arrival point on xy plane of ECI frame