AAS 18-489

AN ANALYTICAL 3D SHAPE-BASED ALGORITHM BASED ON ORBITS INTERPOLATION FOR MULTI-REVOLUTIONS LOW-THRUST TRAJECTORY OPTIMIZATION WITH ECLIPSES AND PERTURBATIONS

J. Prinetto*, M. Lavagna†

A novel 3-dimensional shape based algorithm is proposed in order to extend the domain of analytical solutions to planeto-centric mission scenarios, in which hundreds or thousands of revolutions are required. Due to the strong physical meaning of the shape the method outputs a trajectory close to the real optimal solution. Practical mission constraints are easily formalized, such as maximum thrust threshold and eclipses; moreover, relevant perturbations effects can be considered; free and fixed time of flight are manageable as well. The approach is almost completely analytic, beneficial to significantly lower the computational load, well suited for complex mission scenarios near optimal solutions fast detection.

INTRODUCTION

In the lasts decades Solar Electric Propulsion (SEP) became of primary interest both for long and complex interplanetary missions (such as ESA BepiColombo or NASA Dawn) and for Earth-centered satellite station keeping. The continuous progress on SEP will probably extend the domain of these kind of thrusters to the main propulsion system of planeto-centric missions, such as tug vehicles and launcher's upper stages. On the other hand the low thrust trajectory design and optimization is still a challenge. The optimization of a trajectory using SEP from a mathematical point of view is a continuous optimal control problem of an especially complicated kind:[1] the system of equation governing the dynamics, that is reported in Eq. (1), consists of seven coupled non-linear differential equations (six for the state plus the mass). Perturbations, that are time and state dependent, complicate the problem, as well as the high number of discontinuities that are typically present in a mission (vehicle separation, gravity assists, eclipses). Analytical solutions are available only for unrealistic simple cases.[2–4] Moreover, the structure of the solution in most of the cases is unknown a-priori.[1,5]

$$\begin{cases} \dot{r} = v_r \\ \dot{\theta} = \omega \\ \dot{z} = v_z \\ \dot{v}_r = -\frac{\mu}{\sqrt{(r^2+z^2)^3}} r + r\omega^2 + \frac{T\cos(\beta)\sin(\alpha)}{m} + a^r_{pert} \\ \dot{\omega} = \frac{1}{r}\left(\frac{T\cos(\beta)\cos(\alpha)}{m} - 2\dot{r}\omega + a^t_{pert}\right) \\ \dot{v}_z = -\frac{\mu}{\sqrt{(r^2+z^2)^3}} z + \frac{T\sin(\beta)}{m} + a^z_{pert} \\ \dot{m} = -\frac{T}{I_s g_0} \end{cases} \tag{1}$$

Direct or Indirect optimization methods need typically a good guess in order to converge to an optimal solution, and cannot manage a-priori unknown structure for the solution. Moreover, they require a high computational effort that makes these methods unsuitable for large search spaces.[1,6,7] In order to overcome this shortcut shape-based algorithm have been introduced. Using Shape-based algorithms the continuous

*Politecnico di Milano, Aerospace Science and Technology Department, Via La Masa, 34 20156 Milano - Italy.
†Politecnico di Milano, Aerospace Science and Technology Department, Via La Masa, 34 20156 Milano - Italy.

optimal control problem is reverted: a shape, with proper degrees of freedom, is assumed for the trajectory and the control law is recovered. Typically some assumption, such as the tangential direction of the thrust, are assumed in order to get faster analytical solutions.[8,9] These methods, working on a subspace of the problem, are able to give only a suboptimal solution and are extremely fast if compared with the others.[5,8,10] They are well suited for fast search of sub-optimal solutions in wide search domains using Heuristics algorithms, with the possibility to use multiple/multidisciplinary objectives.[5] These solutions can be used or as initial guests for direct or indirect optimizations or in the earlier phases of the design of a space mission.[5,10] The first shapes developed, was able only to solve simple planar problems without the possibility to impose exact boundary conditions on the positions and velocities and with no TOF constrain.[10] These algorithms were only used to have a quickly estimation of the low thrust trajectory cost and to generate guests trajectories. An important improvement in flexibility of the trajectory and in precision of the solution was proposed by De Pascale and Vasile;[11] the possibility to impose exactly BC gives also the chance to include Gravity assist maneuvers in the trajectory.[12] Conway and Wall developed a new simple but effective shape for both 2D problems and approximated 3D problems with small displacement from plane.[8,9] Other improved shape-based algorithms were found by Novak and Vasile, in which a new coupling between analytic solution and LQR controller is presented.[13] Furier series have also been used in order to generate more effective shapes able to yield solutions nearer to the optimal one.[14] All the above-mentioned shapes, and some other variations proposed by other authors, give very god results in interplanetary trajectory design: the flexibility of the methods and the reduced computational costs allow to design and optimize complex mission scenarios, with mixed integer and continuous variables and with multidisciplinary objective.[5] Unfortunately all the above-mentioned algorithms fail or give unrealistic solutions whenever applied in Earth-Centered mission scenarios. Indeed in this peculiar environment the proximity of the attractor makes the dynamics much more constrained: hundreds or thousands of revolutions are typically needed to move the satellite between two different orbits, and on one revolution the osculating elements remains almost unchanged. Purely geometrical shapes are not able to follow this behavior, especially when eccentric orbits are considered.[8–10] Moreover, eclipses introduces an high number of discontinuities that are not manageable with these algorithms. The purpose of this work is to develop a novel shape-based algorithm able to deal with Earth-centered problems. The working principle of the algorithm is to transform the complex multi-revolution problem in an high number of simpler single revolution trajectories that are solved efficiently with a novel 3D shape-based algorithm. In order to do that a proper number of intermediate orbits is introduced and accurately placed in order not to exceed a threshold on the required thrust. In the first section the developed single revolution algorithm is analyzed, in section 2 the planeto-centric and interplanetary extensions are presented, while section three is devoted to the analysis of some test cases.

SINGLE REVOLUTION ALGORITHM

In this chapter the developed single-revolution shape-based algorithm will be analyzed in-depth and derived in both fixed and free TOF formulations starting from the equation of motion of a thrusting spacecraft via a nonlinear interpolation between departure and target orbits. The working principle of the algorithm is to compute the attractor distance and the declination above the reference plane of the trajectory (respectively s and δ) using a non-linear interpolation between the corresponding functions on the departure and target orbits. The kinematic and the dynamics are then recovered using a semi-analytic procedure. Following the requirements analysis the equation of motion will be parametrized, the non-linear interpolation will be introduced and the architecture of the algorithm will be explained. The most important blocks and equations will be also discussed within this section. Some kinematics relations, even if important in the numerical implementation, are not fundamentals for the comprehension of the work. For shake of simplicity these equations are reported only in appendix.

Requirements

The requirements was selected in order to drive the design of an algorithm able to deal with the highly constrained dynamics that is peculiar of one revolution around the Earth using low thrust propulsion. They are also thought to include as much trajectories as possible, making the tool useful in a wide range of working

conditions. The fundamentals requirements are reported below:

1. The algorithm shall link two different states (Modified Equinoctial Elements[15] or Keplerian Parameters) with the possibility also to impose the time of flight.
2. The algorithm shall work with any couple of orbits that are physically linkable with zero radial thrust[8] (every set of orbits for which Eq. (11) is positive along the whole path) without show singularities, including polar and retrograde orbits.
3. The thrust acceleration shall be exactly recovered via an analytical procedure.
4. The thrust and mass profiles shall be recovered via numerical integration of Tsiolkovsky equation.[16]
5. The required thrust shall tend to zero as the distance between initial and final orbits decreases, independently from eccentricity and departure or arrival anomaly:

$$\lim_{\Delta KP \to 0} \max\left(\frac{|T|}{m}\right) = 0 \qquad \forall \quad e, i, \theta_1, \theta_2 \tag{2}$$

The last requirement (Eq. (2)) is necessary in order to have feasible solutions in Earth-centered mission scenarios: if it is not fulfilled, it is not possible to arbitrary reduce the maximum thrust required simply adding more intermediates orbits.

Parametrization of low thrust equations

The aims of this section is to formalize the parametrization of the 3D equation motion of a thrusting spacecraft in cylindrical coordinates. Differently from the work done by Wall it makes no assumption on small displacement out the plane.[8] The equations of motion presented in the first chapter can be rewritten as shown in Eq.3.[17]

$$\begin{cases} \ddot{r} - r\dot{\theta}^2 = -\frac{\mu}{s^3} r + \frac{T_{IN}}{m} sin\alpha \\ r\ddot{\theta} + 2\dot{r}\dot{\theta} = \frac{T_{IN}}{m} cos\alpha \\ \ddot{z} = -\frac{\mu}{s^3} z + \frac{T_{OUT}}{m} \end{cases} \tag{3}$$

The idea is to parametrize every quantities involved in Eq. (3) as function of the non-dimensional anomaly x introduced in Eq. (4) and then, under some typical assumptions,[9] recover the thrust, mass and time profiles. Physically '$x(t)$' is the angle between the initial position vector and the projection of the spacecraft on the reference plane normalized with the total transfer angle ψ, as can be seen in Figure 1.

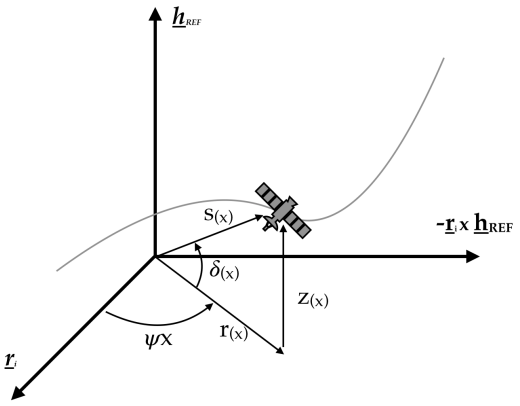

Figure 1. Spacecraft position

$$x(t) = \frac{\theta(t)}{\psi} \qquad with \quad x \in \mathbb{R} \quad \wedge \quad 0 \leq x \leq 1 \tag{4}$$

It is supposed to have also an analytical parametrization of the in-plane radius 'r' and the out-of-plane displacement 'z', as shown in Eq. (5). In general the parametrization of these quantities is not unique: the one selected in this work will be shown later in this section.

$$\begin{cases} r = r(x) \\ z = z(x) \end{cases} \tag{5}$$

The first and second time variation of the in-plane angular displacement can be easily computed as shown in Eq. (6)

$$\begin{cases} \dot{\theta} = \frac{d\theta}{dt} = \psi\dot{x} \\ \ddot{\theta} = \frac{d^2\theta}{dt^2} = \psi\ddot{x} \end{cases} \tag{6}$$

Since both the in-plane radius and the out of plane displacement are considered as function of the in-plane non-dimensional anomaly in order to take their first and second time derivative is necessary to apply the rules for composite function derivation, as shown in Eq. (7)

$$\begin{cases} \dot{r} = \frac{dr}{dt} = \frac{dr}{dx}\frac{dx}{dt} = r'\dot{x} \\ \ddot{r} = \frac{d^2r}{dt^2} = \frac{d}{dt}\left(r'\dot{x}\right) = r''\dot{x}^2 + r'\ddot{x} \\ \dot{z} = \frac{dz}{dt} = \frac{dz}{dx}\frac{dx}{dt} = z'\dot{x} \\ \ddot{z} = \frac{d^2z}{dt^2} = \frac{d}{dt}\left(z'\dot{x}\right) = z''\dot{x}^2 + z'\ddot{x} \end{cases} \tag{7}$$

These quantities can be substituted into Eq. (3) giving Eq. (8).

$$\begin{cases} r''\dot{x}^2 + r'\ddot{x} - r\psi^2\dot{x}^2 = -\frac{\mu}{s^3}r + \frac{T_{IN}}{m}sin\alpha \\ 2\psi r'\dot{x}^2 + r\psi\ddot{x} = \frac{T_{IN}}{m}cos\alpha \\ z''\dot{x}^2 + z'\ddot{x} = -\frac{\mu}{s^3}z + \frac{T_{OUT}}{m} \end{cases} \tag{8}$$

From the first two equations of system shown in Eq. (8), that describe the in-plane motion, is possible to extract the second time derivate of the non-dimensional anomaly, as shown in Eq. (9).

$$\begin{cases} \ddot{x} = \frac{1}{r'}\left[-\frac{\mu}{s^3}r + \frac{T_{IN}}{m}sin\alpha - r''\dot{x}^2 + r\psi^2\dot{x}^2\right] \\ \ddot{x} = \frac{1}{r\psi}\left[-2\psi r'\dot{x}^2 + \frac{T_{IN}}{m}cos\alpha\right] \end{cases} \tag{9}$$

In order to analytically compute the time derivative of 'x' is necessary to remove the dependency from the thrust per unit mass in Eq. (9): this is possible if and only if the in-plane thrust is supposed to be only tangential to the trajectory,[9] indeed in this case the thrust angle is exactly equal to the flight path angle that can be easily computed as shown in Eq. (10).

$$\tan\alpha = \tan\gamma = \frac{v_r}{v_\theta} = \frac{\dot{r}}{r\dot{\theta}} = \frac{r'}{r\psi} \tag{10}$$

Putting together Eq. (10) and Eq. (9) and removing the dependence from the thrust a is possible to compute the square of the time variation of x as shown in Eq. (11).

$$\dot{x}^2 = \frac{\mu r}{s^3\left(r\psi^2 - r'' + 2\frac{r'^2}{r}\right)} = \frac{Nu}{De} \tag{11}$$

Taking the time derivate of Eq. (11) is possible to derive also the second time variation of the non-dimensional anomaly, as shown in Eq. (12).

$$\begin{cases} \ddot{x} = \frac{1}{2}\left(\frac{Nu' - \dot{x}^2De'}{De}\right) \\ Nu' = \mu r' \\ De' = 3\frac{s'}{s}De + s^3\left(r'\psi^2 - r''' + \frac{2rr'r'' - r'^3}{r^2}\right) \end{cases} \tag{12}$$

Since now every time depended quantities are known, it is possible to recover from Eq. (8) the thrust per unit mass as shown in Eq. (13).

$$\begin{cases} \frac{T_{IN}}{m} = \frac{1}{\cos\gamma}\left(2\psi r'\dot{x}^2 + r\psi\ddot{x}\right) \\ \frac{T_{OUT}}{m} = z''\dot{x}^2 + z'\ddot{x} + \frac{\mu}{s^3}z \end{cases} \tag{13}$$

In order to recover the mass profile it is necessary to numerically integrate the Tsiolkovsky equation[18] shown in Eq. (14).

$$\begin{cases} \left|\frac{T}{m}\right| = \sqrt[2]{\left(\frac{T_{IN}}{m}\right)^2 + \left(\frac{T_{OUT}}{m}\right)^2} \\ \frac{dm}{dt} = -\frac{\left|\frac{T}{m}\right|m}{I_S g_0} \end{cases} \tag{14}$$

A numerical integration is also needed to compute the time vector from the variation of the non-dimensional anomaly,[9] as shown in Eq. (15).

$$t = \int_0^t \mathrm{d}\tau = \int_0^t \frac{1}{\dot{x}}\,\mathrm{d}x \tag{15}$$

It is important to underline that, differently from the parametrization proposed by Wall,[8] the one here presented is more general, being suitable for any shape, and removes the assumptions of low displacements from reference plane. The drawback is an increased complexity of the model, as will be shown later in this chapter.

Non-Linear Interpolation

The parametrization of the equation of motion presented before needs a parametric representation of the trajectory in cylindrical coordinates ($r(x)$ and $z(x)$).

These quantities are obtained via a non-linear interpolation between the departure and arrival orbits. This interpolation is performed in spherical coordinates ($s(x)$, $\delta(x)$, ψx, see Figure 2) using an interpolating function $\chi(x)$, as shown in Eq. (16).

$$\begin{cases} s_{(x)} = \left(s_{2(x)} - s_{1(x)}\right)\chi_{(x)} + s_{1(x)} \\ \delta_{(x)} = \left(\delta_{2(x)} - \delta_{1(x)}\right)\chi_{(x)} + \delta_{1(x)} \end{cases} \tag{16}$$

Once the attractor distance and the declination are known it is possible to derive the shape in cylindrical coordinates using Eq. (17).

$$\begin{cases} r = s\cos\delta \\ z = s\sin\delta \end{cases} \tag{17}$$

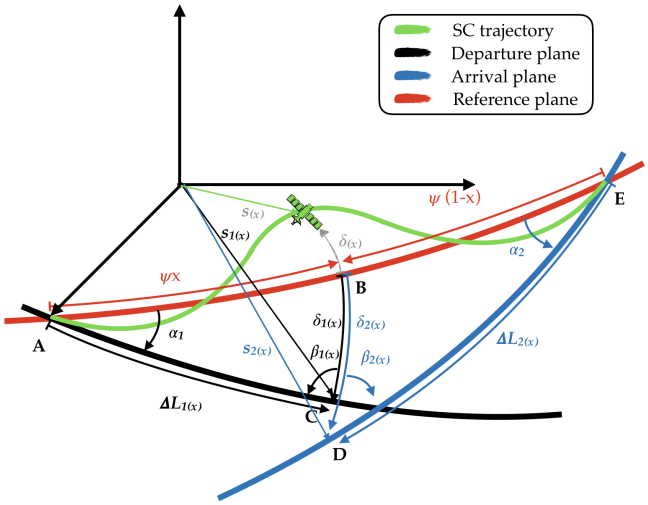

Figure 2. Non-linear interpolation

In order to solve the parametrized equation of motion, it is also necessary to compute the first, second and third derivatives of $r(x)$ and the first two derivatives of $z(x)$ with respect to x as shown in Eq. (18) and Eq. (19) respectively.

$$\begin{cases} r' = -s\delta'\sin\delta + s'\cos\delta \\ r'' = -\left(2s'\delta' + s\delta''\right)\sin\delta + \left(s'' - s\delta'^2\right)\cos\delta \\ r''' = -\left[3\left(s''\delta' + s'\delta''\right) + s\left(\delta''' - \delta'^3\right)\right]\sin\delta + \left[s''' - 3\delta'\left(s'\delta' + s\delta''\right)\right]\cos\delta \end{cases} \tag{18}$$

$$\begin{cases} z' = s\delta' \cos\delta + s' \sin\delta \\ z'' = (2s'\delta' + s\delta'') \cos\delta + \left(s'' - s\delta'^2\right) \sin\delta \end{cases} \tag{19}$$

The derivatives of the spherical coordinates introduced in the last equations can be evaluated using Eq. (20) and Eq. (21).

$$\begin{cases} s' = \Delta s'\chi + \chi'\Delta s + s_1' \\ s'' = \Delta s''\chi + \chi''\Delta s + 2\Delta s'\chi' + s_1'' \\ s''' = \Delta s'''\chi + \chi'''\Delta s + 3\Delta s''\chi' + 3\Delta s'\chi'' + s_1''' \end{cases} \tag{20}$$

$$\begin{cases} \delta' = \Delta\delta'\chi + \chi'\Delta\delta + \delta_1' \\ \delta'' = \Delta\delta''\chi + \chi''\Delta\delta + 2\Delta\delta'\chi' + \delta_1'' \\ \delta''' = \Delta\delta'''\chi + \chi'''\Delta\delta + 3\Delta\delta''\chi' + 3\Delta\delta'\chi'' + \delta_1''' \end{cases} \tag{21}$$

From the previous equations, it is evident the necessity to compute some geometrical quantities of the initial and final orbits: the full set of equations is reported in appendix.

Architecture

The architecture of the algorithm, that is presented in Figure 3, is composed by five main blocks plus two optional, that are activated only if required by the mission.

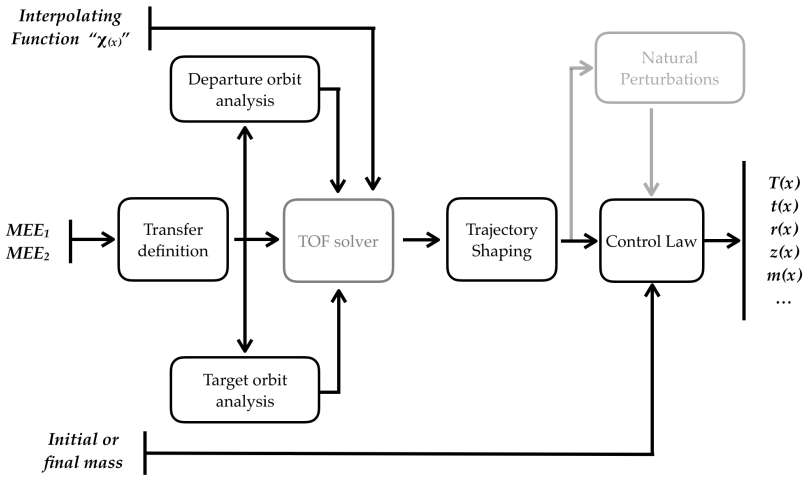

Figure 3. Architecture

A short description of each block it is here reported:

- **Transfer definition:** This block takes as inputs the departure and arrival states. Its aims are to compute the reference frame and the transfer angle. The selected reference frame contains as first axis the initial position and, as third one, the cross product of initial and final radii; the second close the high-handed orthonormal RF. The transfer angle is always considered between π and 3π, in this way singularities due to small transfer angles are avoided. These outputs are the bases of transfers and are therefore passed to the ' Departure/Target' blocks and to the 'Trajectory shaping' block.

- **Departure/Target orbit analysis:** These blocks take as input the data coming from the Transfer definition one. They compute some important geometrical quantities that will be used for the interpolation of the shape. The outputs include all the geometric quantities of interest of the departure and arrival orbits and they are passed directly to the 'Transfer shaping' block.

- **TOF solver:** The aim of this block is to impose a prescribed time of flight to the shape: this block is activated only when TOF constrained problems have to be faced. In order to do that, the shape itself shall contain a further degrees of freedom not fixed with the boundary conditions. This further degrees of freedom is introduced in the interpolating function $\chi(x, a)$. In order to force the time of flight to be equal to the desired one, it is necessary to numerically solve Eq. (22) in the unknown a

$$TOF - \int_0^t \frac{1}{\dot{x}(a)} \, \mathrm{d}x = 0 \tag{22}$$

  this can be easily performed using a Newton algorithm initialized with $a = 0$. The integral presents in Eq. (22) shall be solved numerically at each step of the Newton solver.[19] For this purpose a Cavalieri-Simpson integration scheme is adopted in order to guarantee an acceptable compromise between numerical error reduction and fastness of computation.[19] The function $\dot{x}(a)$ can be computed at each step using Eq. (11). The output of this block is the interpolating function, that is passed to the trajectory shaping block.

- **Trajectory shaping:** This block takes as input the geometries of departure and target orbits computed in the previous blocks and an interpolating function, eventually corrected by the TOF solver block. It computes the exact geometry of the transfer using the non-linear interpolation between departure and target orbits explained before. The output includes the whole geometry and is directly passed to the 'Control law' block and to the 'Natural perturbation' one.

- **Natural perturbations:** This block takes as inputs the trajectory and computes the acceleration due to the natural perturbations (J2 effect). The output, consisting of the acceleration, is sent to the 'Control Law' Block. This block is not fundamental, and can be manually switched on if needed.

- **Control Law:** This block takes as input the geometry of the transfer and the natural perturbation acceleration. The control law, as well as the time array, are computed via a mixed numerical-analytical procedure. In order to compute the time vector, Eq. (15) is integrated using an high order multi-step predictor-corrector scheme (Adams-Bashford-3 Adams-Multon-4[19]), while for the mass profile the same integration scheme is used to integrate Tsiolkovsky equation[16] (Eq. (14)). These integrations can be performed both forward and backward: in this way it is possible to solve both problems with imposed dry or wet mass. The thrust is computed as the element-wise product of the mass and thrust acceleration vectors (eventually corrected with natural perturbations). Since this block presents two ODE to be solved, it is one of the most time consuming. Anyway, it is important to underline that typically an ODE solver spends most of the time to evaluate the function to be integrated but, as can be seen in Eq. (15) and Eq. (14), these information have been yet computed during previous blocks and are therefore available. Moreover, the selection of an accurate integration scheme is necessary to reduce the number of computation nodes, keeping the numerical errors low.

Interpolating functions

A key element of the developed algorithm is the interpolating function χ. As explained before it contains the boundary conditions, therefore it has to respect some requirements, that are briefly described in this section. The boundary conditions to be imposed are simply the initial and final state. As first requirement, the interpolating function $\chi(x)$ shall be continuous with its derivatives till the third order in the domain $[0; 1]$. The boundary conditions are expressed in cylindrical coordinates as follows:

- **Position:** Looking at Eq. (17) and Eq. (16) , it is easy to derive that initial and final conditions on positions (in plane radius $r(x)$ and out of plane displacement $z(x)$) are automatically satisfied if the interpolating function satisfies Eq. (23)

$$\begin{cases} \chi(0) = 0 \\ \chi(1) = 1 \end{cases} \tag{23}$$

- **Velocity:** From the definition of the radial velocity ($\dot{r}$) and the out of plane velocity ($\dot{z}$) in Eq. (7) it is clear that $r', z', \dot{x}$ must match the corresponding quantities of the initial state for $x = 0$ and final state for $x = 1$. If the previous conditions are verified, the boundary conditions on transversal velocity are automatically satisfied, being this velocity defined as $v_r = r\psi\dot{x}$. The requirements on r' and z' at the initial and final point can be directly derived from Eq. (18) and Eq. (19) and are summarized in Eq. (24)

$$\begin{cases} \chi'(0) = 0 \\ \chi'(1) = 0 \end{cases} \tag{24}$$

For the initial and final conditions on $\dot{x}$, it is necessary to look at Eq. (11) in which appears also the second derivatives of r, therefore it is necessary to impose boundary conditions also on r''. From Eq. (18) is possible to derive the conditions on χ shown in Eq. (25)

$$\begin{cases} \chi''(0) = 0 \\ \chi''(1) = 0 \end{cases} \tag{25}$$

In the examples presented in this work the seventh order polynomial function shown in Eq. (26) was selected to solve the free TOF problems, while the eight order polynomial shown in Eq. (27) was selected to solve constrained TOF problems.

$$\chi(x) = -20x^7 + 70x^6 - 84x^5 + 35x^4 \tag{26}$$

$$\chi(x, a) = a_s x^8 - (20 + 4a_s)x^7 + (70 + 6a_s)x^6 - (84 + 4a_s)x^5 + (35 + a_s)x^4 \tag{27}$$

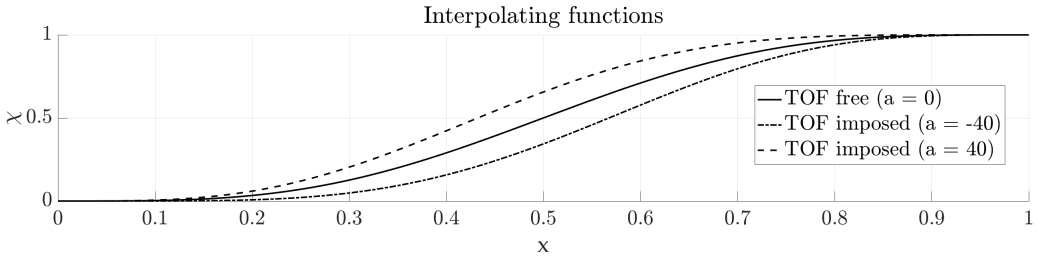

Figure 4. Interp. function comparison

It is easy to prove that these functions satisfy the above-mentioned boundary conditions. For shake of completeness they are plotted in Figure 4 for different values of the parameter a_s. Moreover, it is easy to prove that these interpolation functions lead to solutions with null initial and final thrust: typical undesired peaks at the beginning and the end of the transfer arc (see the thrust profile obtained in[8] and[9]) are avoided. Moreover for $a = 0$ the two functions are identical, and for this reason the TOF solver block above mentioned is always initialized with this value.

On MATLAB R2017 using a laptop equipped with a sixth generation Intel i7 processor at 2.6 GHz this algorithm can evaluate more than 20 thousands revolution per second without imposing TOF and more than 8 thousands with TOF constrained.

MULTI-REVOLUTION APPROACHES

In this section two possible solutions to extend the above-mentioned algorithm to the multi-revolutions trajectories will be presented. The first is the classical way to solve interplanetary trajectories, and it is valid only with a low number of revolutions (no more than 2 or 3), the second, more complicated, is able to deal with thousands of revolutions, including also discontinuities such as eclipses, and therefore it is suitable for planeto-centric mission scenarios.

Interplanetary

Due to the low number of complete revolutions that are typically involved in interplanetary trajectories, one simple but effective possibility is to consider an 'augmented' transfer angle as show in Eq. (28).[8,9]

$$\psi = \psi + 2\pi N_{rev} \tag{28}$$

The solution is not well suited for N larger that 2 or 3, otherwise the shape of the interpolating function generates solutions with a bad distribution of the thrust peaks. This solution is extremely fast since it requires the evaluation of only one trajectory.

Interplanetary trajectories design and optimization introduces some other issues and constrains/objectives that are necessary to identify and formalize. Regardless from the technology adopted, the thrust is linearly related with the power available: typically a specific power between $15\ \frac{W}{mN}$ and $40\ \frac{W}{mN}$ is needed. By definition in interplanetary trajectories the variation of the distance from sun is not negligible and, in most of the cases, it can affect the available power of around one order of magnitude if solar arrays are used as primary power sources due to the dependence of the solar flux to the inverse of the square of the distance.[20]

Solar panels are affected by aging effects that reduces the amount of available power during time. This aging effect can be estimated between 2% and 4%[21] depending on the technology adopter for them. Interplanetary missions can lasts decades[5],[22] therefore the amount of available power is also function of the total time of flight. These effects can be merged together in a unique constrain that, for shake of fastness in the convergence of the optimizer and flexibility in the mission design, is threated as an objective. The physical quantity that synthesize all the above-mentioned issues is obviously the solar panel surface needed to accomplish the mission. It can be performed evaluating Eq. (29)[20] in each computational nodes and considering the maximum value.

$$A_{SA}(x) = \frac{kT + P_{ss}}{\eta_{tot} \cos\phi \left(1-\beta\right)^{t\ \frac{\phi_{Earth}}{s^2}}} \tag{29}$$

In Eq. (29) k is the power per unit thrust, P_{ss} is the power consumption of the rest of the spacecraft, η is the efficiency of the power production/conversion, $cos(\phi)$ is the cosine of the sun angle, β is the aging factor of the solar panels and ϕ_{Earth} is the solar flux at 1 $[AU]$. The sun-angle can be supposed a-priori or computed point by point from the control law, if the geometry of the spacecraft is known. Since the optimal solution is the same for every positive multiple of the objective function, Eq. (29) can be simplified in order to generate an objective function that is less dependent from the specific parameter of the spacecraft giving the objective function in Eq. (30).

$$obj = MAX\left[\frac{T(x)s(x)^2}{\left(1-\beta\right)^{t(x)}}\right] \tag{30}$$

This objective function, that due to the presence of the aging effect tries to contain also the Time of Flight, can be used[5] inside a multi-objective multidisciplinary genetic algorithm together with the fuel mass. It is possible also to impose a constrain on the maximum thrust or introduce it as a third objective function: in this last case the Pareto front allows to select directly the solution that better fits the requirements and the constrains of the mission. In the following paragraph two different approaches are reported.

Planeto-centric

As introduced at the beginning of this work, the basic principle to solve this kind of mission scenarios is to introduce some intermediate keplerian orbits. Within this framework it is possible, as will be analyzed later in this section, to switch off thrusters during the eclipses. Before introducing the architecture scheme, it is important to formalize the problem: the initial and final states are imposed as Modified Equinoctial Elements (MEE), as shown in Eq. (31).

$$\begin{cases} MEE_i = [p_i, f_i, g_i, h_i, k_i, L_i] \\ MEE_f = [p_f, f_f, g_f, h_f, k_f, L_f] \end{cases} \tag{31}$$

Since the algorithm can work both forward and backward in time, two different but similar formulations are available: for shake of brevity only the forward algorithm is deeply analyzed. The steps involved are the following:

1. **initialization of the problem:** at the first step of the algorithm the spacecraft is at the initial state, while the desired is the final one, therefore Eq. (32) holds.

$$\begin{cases} MEE_1 = MEE_i \\ MEE_2 = MEE_f \\ m_k(0) = M_{initial} \\ t_k(0) = 0 \end{cases} \tag{32}$$

   with $k = 1$ since the first trajectory has to be designed.

2. **Positioning the k-th intermediate orbit:** the spacecraft is in the position described by MEE_1 with mass $m_k(0)$ and the time after departure in $t_k(0)$. The objective is to place the k^{th} intermediate orbit in such a way that the maximum thrust required is equal to the maximum available one. Accordingly to Eq. (33) the position of the intermediate orbit depends on the value of the parameter η_k: the higher is η_k the higher will be the gap between the current orbit and the intermediate one and so the higher will be the thrust required.

$$\begin{cases} p_k = (p_2 - p_1)\,\eta_k + p_1 \\ f_k = (f_2 - f_1)\,\eta_k + f_1 \\ g_k = (g_2 - g_1)\,\eta_k + g_1 \\ h_k = (h_2 - h_1)\,\eta_k + h_1 \\ k_k = (k_2 - k_1)\,\eta_k + k_1 \end{cases} \tag{33}$$

   The exact position on the intermediate orbit, described by the sixth parameter L_k, is depending on the other five Modified Equinoctial Elements computed in Eq. (33) since, as said before, it must coincide with the entrance on the eclipse, no matter if it is real or fictitious. The model adopted for the eclipse is the standard cylindrical one. It is important to underline that, being the shadowed region function of the time due to the Earth motion around the sun, it has to be computed at each k^{th} step. The desired value of η_k is found solving numerically Eq. (34) in which the term $max\,(T(\eta_k))$ is the maximum thrust required during the k^{th} trajectory that can be computed using the TOF free algorithm presented in the previous section.

$$max\,(T(\eta_k)) - T_{available} = 0 \tag{34}$$

   The numerical solution of Eq. (34) in general is not straightforward since there are points in which its continuity is not guaranteed, as well as the existence of the solution itself. The adopted solution is an hybrid Newton-Bisection algorithm developed ad-hoc. As first step, the algorithm tries to solve the equation with Newton method, then if it fails, the algorithm tries again with the Bisection method. If also in this case the algorithm fails in reaching a predefined tolerance in a certain number of iterations, the equation is transformed into an inequality, as shown in Eq. (35).

$$max\,(T(\eta_k)) - T_{available} < 0 \tag{35}$$

   The last inequality can always be solved since the developed shape-based algorithm fulfill the requirement in Eq. (36)

$$\lim_{\Delta MEE \to 0} \max\left(\frac{|T|}{m}\right) = \lim_{\eta_k \to 0} \max\left(\frac{|T|}{m}\right) = 0 \tag{36}$$

If the value of η_k found with previous equations is equal or greater than one it means that the thrust available on-board is sufficient to reach the final position MEE_2, as can be seen from Eq. (33): in this case η_k is automatically switched to one and the k^{th} trajectory is computed again; the algorithm stops. Otherwise if η_k is between 0 and 1 it is necessary to prepare the states for the next step using Eq. (37): the new starting position is the exit from the eclipse of the k^{th} intermediate orbit, the initial mass of the spacecraft on the trajectory $k+1$ is exactly equal to the final mass of the trajectory k since during the eclipse no fuel is consumed. The time after departure at the beginning of the $k+1$ trajectory is equal to the arrival time of the trajectory k plus the time spent in shadow ($\Delta t_{eclipse}$)

$$\begin{cases} MEE_1 = MEE_k \\ MEE_2 = MEE_f \\ m_{k+1}(0) = m_k(1) \\ t_{k+1}(0) = t_k(1) + \Delta t_{eclipse} \end{cases} \tag{37}$$

The value of k is then increased and the algorithm goes back to the beginning of point 2. The cycle stops when η_k is equal or greater to one.

3. **Trajectory analysis**: For fastness purposes, the previous block gives as outputs only the initial/final mass and the time of flight. If more information are needed once the intermediate orbits are placed, this block is activated and compute all the kinematics and dynamics quantities of the trajectory. The computational time required by the algorithm is in around 50% higher if this block takes place, therefore in the optimization of complex mission scenarios it is better to run it only on a limited number of solutions.

The backward version of the algorithm involves the same steps, with some important differences:

- Now the initial state is the arrival one, and the desired is the departure one, therefore Eq. (32) is substituted with Eq. (38).

$$\begin{cases} MEE_1 = MEE_f \\ MEE_2 = MEE_i \\ m_k(1) = M_{final} \\ t_k(1) = 0 \end{cases} \tag{38}$$

- The mass imposed in Eq. (38) is the final one instead of the initial one and the variable $t_k(x)$ now means 'time before the arrive' instead of 'time after departure' ant it is a negative quantity.
- The k^{th} intermediate orbit is integrated itself backward in time, therefore the update of the states becomes the one shown in Eq. (39) instead of the one in Eq. (37).

$$\begin{cases} MEE_1 = MEE_i \\ MEE_2 = MEE_k \\ m_{k+1}(1) = m_k(0) \\ t_{k+1}(1) = t_k(0) - \Delta t_{eclipse} \end{cases} \tag{39}$$

It means that the final mass to be imposed to the $k^{th}+1$ single revolution trajectory is the initial one computed for the k^{th} revolution, as happens for the time.

The effect of having both the forward and backward direction for the trajectory design allows to keep into account more scenarios. The high level block scheme of the algorithm is presented in Figure 6 for shake of completeness. Figure 5 shows how the algorithm works for a geocentric (unfeasible) trajectory.

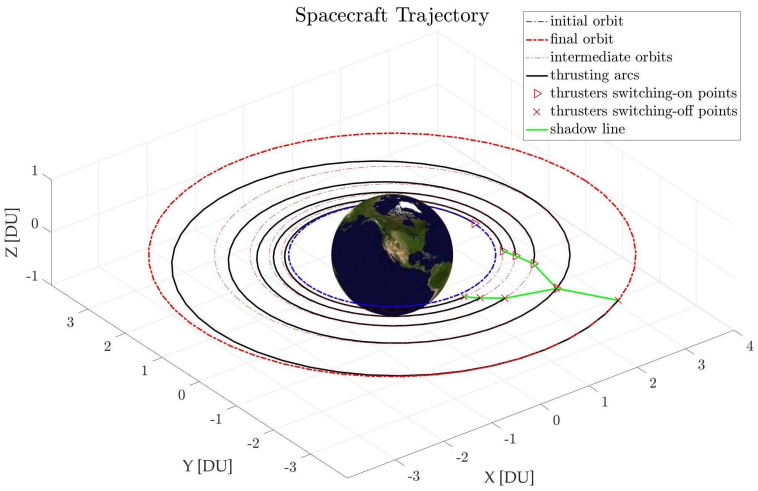

Figure 5. Example of multi-revolution application

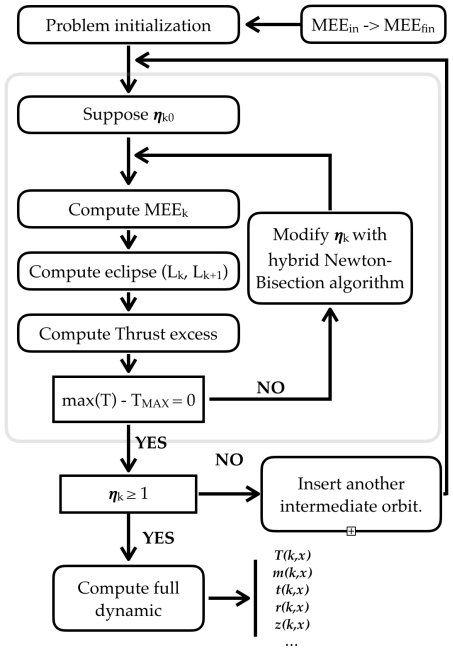

Figure 6. Multi-revolution algorithm architecture

TEST CASES

In this section some test cases are presented. All the computations have been performed with MATLAB R2017b on an laptop equipped with a sixth generation Intel i7 processor working at 2.6 GHz without adopting parallel computation.

Electric Orbital Rising to GEO

In this scenario the possibility to launch a satellite in GEO using electric propulsion is explored. The satellite, with a dry mass of 800 [kg], specific impulse of 3800 [s] and max. thrust of 0.5 [N], is firstly placed in a parking orbit by the European VEGA launcher,[23] and then with its own electric propulsion unit reaches GEO.

The inclination of the parking orbit is fixed at $5.4\,[deg]$, the minimum reachable from Kourou without a plane change, and the standard parking orbit plane for VEGA.[23] The apocenter and pericenter radii are considered as degree of freedom for the optimization process. Their values can range from $1.03\,[DU]$ to $6.6108\,[DU]$, including therefore any possible intermediate orbit between LEO and GEO. Due to the inclination of the Earth rotation axis, eclipses encountered by a satellite above LEO orbits are strongly affected by the period of the year: a satellite in GEO goes in Earth shadow only nearby the equinoxes.[17] Since in this example the spacecraft is supposed to be able to thrust only in sunlight, the solution will depend also on the season of the GEO arrival, therefore the two opposite cases (arrival at the equinoxes or at the solstices) have been analyzed. In order to highlight the importance of having included the eclipses in the model also the solution without them will be reported.

The most significant scenario is reported in Figure 7. The GEO injection epoch is nearby the Solstice, and an impulsive disposal maneuver is included in the optimization process in order to reenter the upper stage 'AVUM' in the atmosphere. Red lines represent the launchable mass using VEGA launcher: since a complete set of information for the launcher is not available, data are extrapolated applying the Tsiolkovsky equation to the last stage of the launcher (AVUM) from the reference orbit available on the user-manual.[23] The time optimal problem was solved for all the above-mentioned cases using a Nelder-Mead simplex algorithm[24,19] modified with a penalty method in order to force the solution to show an initial mass lower than the launchable one on the same orbit. The decision to adopt a derivative-free algorithm arises from the fact that, being the number of revolution discrete, the time of flight is not continue.

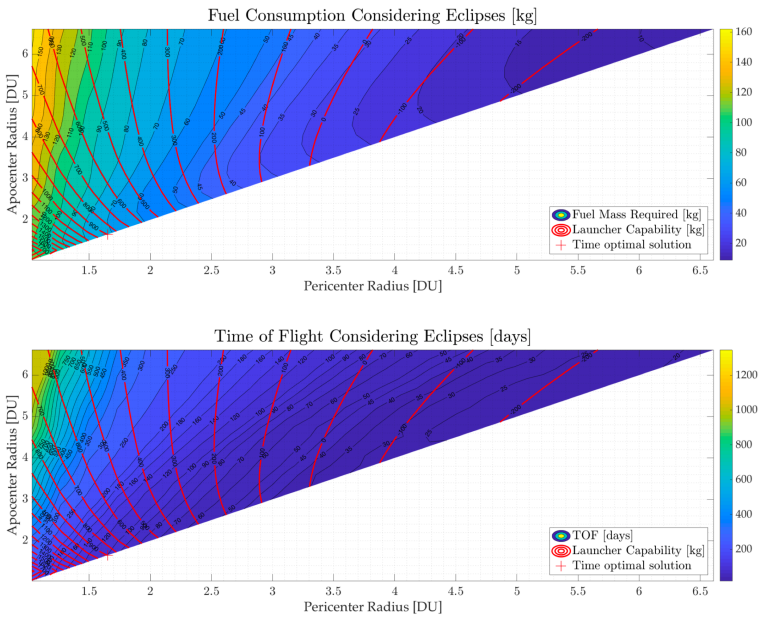

Figure 7. Orbit rising to GEO

Table 1. Time optimal constrained solutions

Model	Eclipses		Eclipses and disposal	No eclipses
Arrival epoch	Equinox	Solstice	Solstice	—-
Fuel Mass [kg]	62.30	61.72	71.5	61.59
TOF [days]	83.89	79.84	98.47	66.5
Revolutions [-]	280	279	425	214
r_p parking [DU]	1.8994	1.9019	1.6517	1.8816
r_a parking [DU]	1.9030	1.9029	1.6524	1.9207
CPU time [s]	23	31	41	12

Table 1 shows different solutions for similar scenarios, changing or the arrival epoch, removing the eclipses or avoiding the disposal maneuver. The fuel consumption is similar for all the trajectories while the time of flight increases between 20% and 25% if eclipses are considered. The effect of the injection season is much smaller: this is due to the fact that, even if nearby the equinoxes eclipses are present at any distance from ground, the fraction of time spent in shadow decreases with the radius. The CPU time highlights the capability of the algorithm to find sub-optimal solutions for multi-revolutions discontinuous trajectories very quickly:

in these simulations 50 computational nodes per revolution have been used for the optimization processes and 100 for the plot of the final trajectory. All optimization processes have been initialized with the reference VEGA parking orbit (200 $[km]$ x 1500 $[km]$ height LEO orbits), the differences in the CPU times between the solutions reflect the different number of revolutions required and the increase in complexity if eclipses are encountered. In literature there is no database with time optimal solutions of GEO rising problems including eclipses in the model, therefore the only crosscheck can be done with the solutions without them.

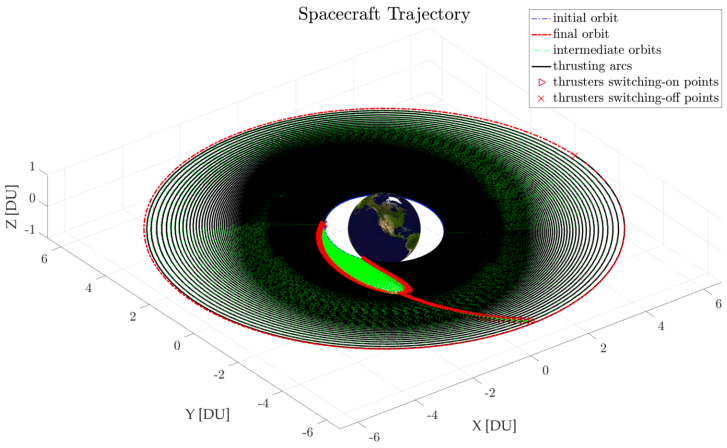

Figure 8. Time optimal trajectory

The optimal solution found within this work is slightly more expensive if compared with the ones available in literature, but the CPU times are order of magnitude lower[25].[26] For shake of completeness in Figure 8 is reported the 3D trajectory, with a quasi circular switching orbits with a radius of 2 [DU], for the case with eclipse and disposal maneuver that represents the most complex solution among the ones listed in Table 1. It is easy to see that the distribution of the density of the intermediate orbits is much higher near the Earth. Regarding the eclipses, it is possible to see the distortion of the shadow region caused by the motion of the Earth around the Sun and, as lower effect, the plane changes.

Earth-Mars Rendezvous

The Earth-Mars rendezvous problem is a classical scenario for the validation of low thrust algorithms.[11]

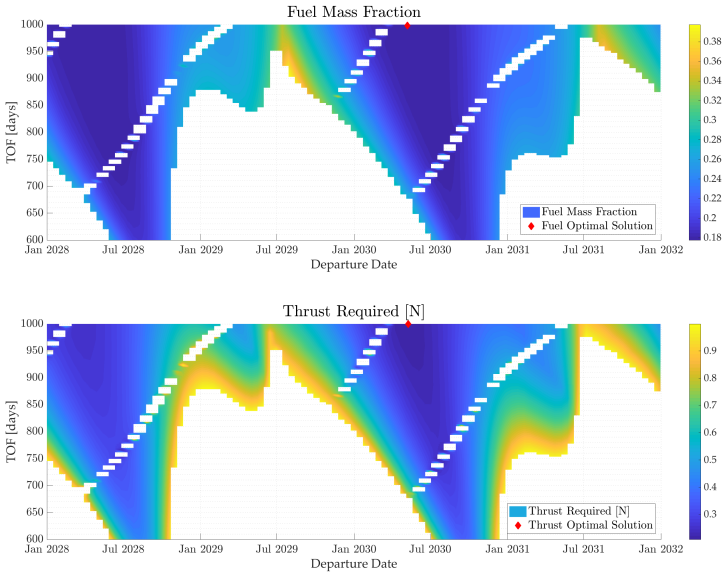

Figure 9. Launch opportunities

The spacecraft has a dry mass of 1000 [kg], a specific impulse of 3000 [s] and a maximum thrust of 0.22 [N] The thrust required and the fuel mass fraction over the whole search domain are reported in Figure 9; white regions are the ones in which or the thrust exceeds 1 $[N]$ or the fuel mass fraction exceeds 0.5 $[-]$. The thrust optimal and fuel mass fraction optimal solutions are also reported: it is evident that for this algorithm the search domain proposed by Vasile is too small and therefore both the optimal solutions are located at the border of the domain. Anyway, inside the domain it is possible to recognize two convenient regions in which the thrust and the fuel mass fraction are low even for Time of Flight reduction. The fuel optimal solution is similar to the one found by Vasile and De Pascale, with the same fuel mass fraction of 0.177 [-] and a slightly higher time of flight of 1000 days ; the departure date is the 8^{th}

of May 2030. The CPU time, using the standard MATLAB genetic algorithm with a population of 100 individuals and stopping criteria based on the average change of the cost function, is lower than 10 seconds.

Earth-Nereus Rendezvous

This scenario was selected to underline the ability of the shape based algorithm to find near optimal solution when high elliptical orbits are considered. Nereus is a Near Earth Object with an high elliptical orbit on a plane slightly different with respect to the Earth one. Its pericenter is located nearby the Earth's one, while the apocenter is at 1.5 [AU], therefore a quasi ballistic solution with a non zero escape velocity is expected if or the spacecraft's fuel mass fraction or the thrust required are selected as objectives. In order to try to find the best trajectory, an extremely wide search space is considered: the degrees of freedom and their ranges are reported in Table 2.

Table 2. Search domain

	Range	Optimum
Departure Date	from 2030 to 2050	9^{th} February 2042
Time of Flight [days]	from 500 to 1500	690.5
Number of revolutions [-]	from 0 to 2	1
v_{inf} departure [km/s]	from 0 to 6	5.93
v_{inf} in plane angle [deg]	from -90 to +90	-8.07
v_{inf} out of plane angle [deg]	from -90 to +90	44.79

Regarding the optimization process, the MATLAB genetic algorithm is adopted with a population of 1000 individuals and the optimal trajectory is obtained after 5 minutes with 100 computational nodes.

As can be seen from the output trajectory reported in Figure 10, the launcher inserts the spacecraft, that has a mass of 1000 kg and a specific impulse of 3000 s, directly in a quasi ballistic orbit, as expected from theory: the only use of the thrusters is the relative approach phase to Nereus; slightly more than 10 mN of thrust are sufficient, and the fuel mass fraction required is only 0.0052. This example shows that the developed algorithm is capable to manage also high elliptical orbits in interplanetary trajectories: this is possible only because the peculiarity of the shape is to be a non-linear interpolation between arrival and departure orbits.

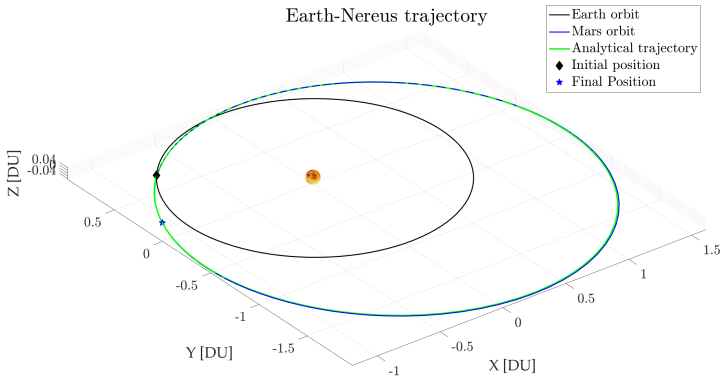

Figure 10. Earth Nereus trajectory

CONCLUSION

The goal to develop a shape-based algorithm able to deal with planeto-centric scenarios is achieved. More in detail the algorithm shows a good convergence and it is extremely fast and sufficiently flexible for the purposes of this work. The strategy developed to introduce eclipses is fast and efficient, and can work with any type of orbits without singularities. The algorithm converges for any value of the maximum thrust if purely keplerian motion is considered. The speed of the algorithm is guaranteed by the fact that, even if an elevate number of expression have to be evaluated, they are all composed by the same block of repeated terms, that can be computed only one time. Moreover, most of the trigonometrical expression are not directly

evaluated by MATLAB functions, but they are evaluated using the trigonometrical relations. Unfortunately, the strategy adopted to take into account perturbation is not so effective: they are simply zeroed by the thrust. Unfortunately, in Geocentric environment, especially in LEO, perturbations can be of the same order of magnitude of the control thrust, and therefore the algorithm doesn't converge for these levels of thrust. As future work, it is planned to introduce directly perturbations in the intermediates orbits, indeed by doing that the interpolated shapes contain the perturbation effect, and will be able to follow closer the perturbed dynamics, giving the possibilities to use also a control thrust lower with respect to the perturbation forces.

NOTATION

a	Semi-Major axis
a_{pert}	Perturbation acceleration
a_s	Semi-Major axis
A_{sa}	Solar panels surface
e	Eccentricity
f	Second MEE
G_{loss}	Gravity loss
g	Third MEE
h	Fourth MEE
I_s	Specific Impulse
i	Inclination
k	Fifth MEE
L	Sixth MEE
m	Mass
N_{rev}	Number of revolution
P_{SS}	Spacecraft Power consumption
p	First MEE
r	In-plane attractor distance
s	attractor distance
T	Thrust
T_{IN}	In-plane Thrust
T_{OUT}	Out of plane Thrust
t	Time
v_r	Radial velocity
v_z	Normal velocity
v_θ	transversal velocity
x	Non-dimensional anomaly
z	Out of plane displacement
α	In-plane thrust angle
β	Out of plane thrust angle
γ	Flight path angle
δ	Declination
η_{tot}	Power production efficiency
θ	Anomaly
ϕ	Solar aspect angle
χ	Interpolating function
ψ	Total transfer angle
ω	angular velocity

APPENDIX A: LIST OF DERIVATIVES

In this appendix the equation that are fundamental to the geometrical interpolation of the trajectory are reported. The subscript '1' indicates the initial orbit, while the subscript '2' indicates the arrival one.

Departure orbit

The inclination of the initial orbit with respect to the reference plane can be computed as in Eq. (40).

$$\begin{cases} \cos\alpha_1 = \hat{\mathbf{h}}_1 \cdot \hat{\mathbf{h}}_{REF} \\ \sin\alpha_1 = \xi_1\sqrt{1-\cos\alpha_1{}^2} \\ \begin{cases} \xi_1 = 1 & if \quad \mathbf{v}_i \cdot \hat{\mathbf{h}}_{REF} > 0 \\ \xi_1 = -1 & if \quad \mathbf{v}_i \cdot \hat{\mathbf{h}}_{REF} < 0 \end{cases} \end{cases} \tag{40}$$

The declination $(\delta(x)_1)$ of the initial orbit over the reference plane can be computed using Eq. (41), while its derivatives can be computed using Eq. (42), Eq. (43) and Eq. (44).

$$\sin\delta_1 = \sin\alpha_1 \frac{\sin(\psi x)}{\sin\beta_1} \tag{41}$$

$$\delta_1' = \frac{\psi \sin\alpha_1 \cos(\psi x) - \beta_1' \cos\beta_1 \sin\delta_1}{\cos\delta_1 \sin\beta_1} \tag{42}$$

$$\delta_1'' = \frac{-\psi^2 \sin\alpha_1 \sin(\psi x) + \sin\beta_1 \sin\delta_1 \left(\delta_1'^2 + \beta_1'^2\right)}{\cos\delta_1 \sin\beta_1} + \\ - \frac{2\delta_1'\beta_1' \cos\delta_1 \cos\beta_1 + \beta_1'' \sin\delta_1 \cos\beta_1}{\cos\delta_1 \sin\beta_1} \tag{43}$$

$$\delta_1''' = \frac{-\psi^3 \sin\alpha_1 \cos(\psi x) + \sin\beta_1 \sin\delta_1 \left(3\delta_1'\delta_1'' + 3\beta_1'\beta_1''\right)}{\cos\delta_1 \sin\beta_1} + \\ + \frac{-\cos\beta_1 \cos\delta_1 \left(3\delta_1'\delta_1'' + 3\beta_1'\beta_1''\right) + \cos\beta_1 \sin\delta_1 \left(3\delta_1'^2\beta_1' + \beta_1'^3 - \beta_1'''\right)}{\cos\delta_1 \sin\beta_1} + \\ + \frac{\sin\beta_1 \cos\delta_1 \left(3\beta_1'^2\delta_1' + \delta_1'^3\right)}{\cos\delta_1 \sin\beta_1} \tag{44}$$

In the previous equations another spherical angle ($\beta(x)_1$ in Figure 2) is introduced together with its derivatives. They can be computed using Eq. (45).

$$\begin{cases} \beta_1 = \arccos\left(\sin\alpha_1 \cos(\psi x)\right) \\ \beta_1' = \psi \sin\alpha_1 \frac{\sin(\psi x)}{\sin\beta_1} \\ \beta_1'' = \frac{\psi^2 \sin\alpha_1 \cos(\psi x) - \cos\beta_1 \beta_1'^2}{\sin\beta_1} \\ \beta_1''' = \frac{-\psi^3 \sin\alpha_1 \sin(\psi x) - 3\beta_1'\beta_1'' \cos\beta_1 + \beta_1'^3 \sin\beta_1}{\sin\beta_1} \end{cases} \tag{45}$$

The angle $\Delta L_1(x)$ is fundamental to compute the Longitude ($6^{th} MEE$) on the initial orbit at each x and so the attractor distance of the departure orbit as function of x; it can be computed with Eq. 46 and Eq. 47.

$$\begin{cases} \sin(\Delta L_1) = \frac{1}{\sin\alpha_1} \sin\delta_1 \\ \cos(\Delta L_1) = \cos(\psi x) \cos\delta_1 \end{cases} \tag{46}$$

$$\begin{cases} \Delta L_1' = \frac{\delta_1'}{\sin\alpha_1 \cos(\psi x)} \\ \Delta L_1'' = \frac{\delta_1'' + \psi \sin\alpha_1 \sin(\psi x) \Delta L_1'}{\sin\alpha_1 \cos(\psi x)} \\ \Delta L_1''' = \frac{\delta_1''' + 2\psi \sin\alpha_1 \sin(\psi x) \Delta L_1'' + \psi^2 \sin\alpha_1 \cos(\psi x) \Delta L_1'}{\sin\alpha_1 \cos(\psi x)} \end{cases} \tag{47}$$

Target orbit

The inclination of the arrival orbit with respect to the reference plane can be computed as in Eq. (48).

$$\begin{cases} \cos\alpha_2 = \hat{\mathbf{h}}_2 \cdot \hat{\mathbf{h}}_{REF} \\ \sin\alpha_2 = \xi_2\sqrt{1-\cos\alpha_2{}^2} \\ \quad \xi_2 = 1 \qquad if \quad \mathbf{v}_f \cdot \hat{\mathbf{h}}_{REF} < 0 \\ \quad \xi_2 = -1 \qquad if \quad \mathbf{v}_f \cdot \hat{\mathbf{h}}_{REF} > 0 \end{cases} \tag{48}$$

The declination ($\delta(x)_2$) of the arrival orbit over the reference plane can be computed using Eq. (49), while its derivatives can be computed using Eq. (50), Eq. (51) and Eq. (52)

$$\sin\delta_2 = \sin\alpha_2 \frac{\sin(\psi(1-x))}{\sin\beta_2} \tag{49}$$

$$\delta_2' = \frac{-\psi\sin\alpha_2\cos\left(\psi(1-x)\right) - \beta_2'\cos\beta_2\sin\delta_2}{\cos\delta_2\sin\beta_2} \tag{50}$$

$$\begin{aligned}\delta_2'' = \frac{-\psi^2\sin\alpha_2\sin\left(\psi(1-x)\right) + \sin\beta_2\sin\delta_2\left(\delta_2'^2+\beta_2'^2\right)}{\cos\delta_2\sin\beta_2} + \\ - \frac{2\delta_2'\beta_2'\cos\delta_2\cos\beta_2 + \beta_2''\sin\delta_2\cos\beta_2}{\cos\delta_2\sin\beta_2}\end{aligned} \tag{51}$$

$$\begin{aligned}\delta_2''' = \frac{\psi^3\sin\alpha_2\cos\left(\psi(1-x)\right) + \sin\beta_2\sin\delta_2\left(3\delta_2'\delta_2'' + 3\beta_2'\beta_2''\right)}{\cos\delta_2\sin\beta_2} + \\ + \frac{-\cos\beta_2\cos\delta_2\left(3\delta_2'\delta_2'' + 3\beta_2'\beta_2''\right) + \cos\beta_2\sin\delta_2\left(3\delta_2'^2\beta_2' + \beta_2'^3 - \beta_2'''\right)}{\cos\delta_2\sin\beta_2} + \\ + \frac{\sin\beta_2\cos\delta_2\left(3\beta_2'^2\delta_2' + \delta_2'^3\right)}{\cos\delta_2\sin\beta_2}\end{aligned} \tag{52}$$

In the previous equations another spherical angle ($\beta(x)_2$ in Figure 2) is introduced together with its derivatives. They can be computed using Eq. (53).

$$\begin{cases} \beta_2 = \arccos\left(\sin\alpha_2\cos(\psi(1-x))\right) \\ \beta_2' = -\psi\sin\alpha_2\frac{\sin\left(\psi(1-x)\right)}{\sin\beta_2} \\ \beta_2'' = \frac{-\psi^2\sin\alpha_2\cos\left(\psi(1-x)\right) - \cos\beta_2\beta_2'^2}{\sin\beta_2} \\ \beta_2''' = \frac{\psi^3\sin\alpha_2\sin\left(\psi(1-x)\right) - 3\beta_2'\beta_2''\cos\beta_2 + \beta_2'^3\sin\beta_2}{\sin\beta_2} \end{cases} \tag{53}$$

The angle $\Delta L_2(x)$ is fundamental to compute the Longitude ($6^{th} MEE$) on the arrival orbit at each x and so the attractor distance of the arrival orbit as function of x; it can be computed with Eq. 54 and Eq. 55.

$$\begin{cases} \sin\left(\Delta L_2\right) = \frac{1}{\sin\alpha_2}\sin\delta_2 \\ \cos\left(\Delta L_2\right) = \cos\left(\psi(1-x)\right)\cos\delta_2 \end{cases} \tag{54}$$

$$\begin{cases} \Delta L_2' = \frac{\delta_2'}{\sin\alpha_2\cos\left(\psi(1-x)\right)} \\ \Delta L_2'' = \frac{\delta_2'' - \psi\sin\alpha_2\sin\left(\psi(1-x)\right)\Delta L_2'}{\sin\alpha_2\cos\left(\psi(1-x)\right)} \\ \Delta L_2''' = \frac{\delta_2''' + 2\psi\sin\alpha_2\sin\left(\psi(1-x)\right)\Delta L_2'' + \psi^2\sin\alpha_2\cos\left(\psi(1-x)\right)\Delta L_2'}{\sin\alpha_2\cos\left(\psi(1-x)\right)} \end{cases} \tag{55}$$

Attractor distances

In order to compute the attractor distance the first step is to compute the Longitude at each position x on the initial and final orbits using Eq. (57).

$$\begin{cases} l_1(x) = L_1 + \Delta L_1(x) \\ l_1'(x) = \Delta L_1'(x) \end{cases} \tag{56}$$

$$\begin{cases} l_2(x) = L_2 - \Delta L_2(x) \\ l_2'(x) = -\Delta L_2'(x) \end{cases} \tag{57}$$

The attractor distance on the initial and final orbit can be computed using Eq. (58).

$$\begin{cases} s_i(x) = \frac{p_i}{q_i(x)} \\ s_i(x)' = -\frac{p_i q_i'}{q_i^2} \\ s_i(x)'' = 2\frac{p_i q_i'^2}{q_i^3} - \frac{p_i q_i''}{q_i^2} \\ s_i(x)''' = -6\frac{p_i q_i'^3}{q_i^4} + 6\frac{p_i q_i' q_i''}{q_i^3} - \frac{p_i q_i'''}{q_i^2} \end{cases} \tag{58}$$

In which another term appears with its derivatives; it can be computed using Eq. (59)

$$\begin{cases} q_i(x) = 1 + f_i \cos l_i(x) + g_i \sin l_i(x) \\ q_i(x)' = (-f_i \sin l_i + g_i \cos l_i)\, \Delta L_i' \\ q_i(x)'' = (1 - q_i)\Delta L_i'^2 + q_i' \frac{\Delta L_i''}{\Delta L_i'^2} \\ q_i(x)''' = -q_i' \Delta L_i'^2 + 3(1 - q_i)\Delta L_i' \Delta L_i'' + q_i' \frac{\Delta L_i'''}{\Delta L_i'} \end{cases} \tag{59}$$

REFERENCES

[1] B. A. Conway, *Spacecraft Trajectory Optimization*, ch. 3. Cambridge University Press, 2010.

[2] T. H.S., "Take-Off from Satellite Orbit," *Journal of the American Rocket Society*, Vol. 23, No. 4, 1953, pp. 233–236.

[3] R. H. Battin, ed., *An Introduction to the Mathematics and Methods of Astrodynamics*. Reston: AIAA Education Series, 1999.

[4] Q. A. Mengali G., "Escape from Elliptic Orbit Using Constant Radial Thrust," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 3, 2009, pp. 1018, 1022.

[5] M. L. J. Prinetto, "Main Belt Active Asteroids Samples Collection And Return Mission Design," *Italian Association of Aeronautics and Astronautics XXIV International Conference*, Palermo-Enna, Italy, AIDAA, 18-22 September 2017.

[6] S. R. O. Craig A. Kluever, "Direct Approach for Computing Near-Optimal Low-Thrust Earth-Orbit Transfers," *Journal of Spacecraft and Rockets*, Vol. 35, No. 6, 1998, pp. 509, 515.

[7] C. F.Topputo, "Survey of Direct Transcription for Low-Thrust Space Trajectory Optimization with Applications," *Abstract and Applied Analysis*, Vol. 2014, 2014, p. 15.

[8] B. J. Wall, "Shape-Based Approximation Method for Low-Thrust Trajectory Optimization," *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Honolulu, Hawaii, AIAA, 2008.

[9] B. A. Bradley J.Wall, "Shape-Based Approach to Low-Thrust Rendezvous Trajectory Design," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 1, 2009, pp. 95, 101, 10.2514/1.36848.

[10] J. A. S. Anastassios E. Petropoulos, "A Review of Some Exact Solutions to the Planar Equations of Motion of a Thrusting Spacecraft," *In Proceedings of the 2nd International Symposium on Low Thrust Trajectories*, Toulouse, France, 18 June 2002.

[11] M. V. P. De Pascale, "Preliminary Design of Low-Thrust Multiple Gravity-Assist Trajectories," *Journal of Spacecraft and Rockets*, Vol. 43, No. 5, 2006, pp. 1065, 1076.

[12] J. M. Anastassios E.Petropoulos, "Shape-Based Algorithm for Automated Design of Low-Thrust. Gravity-Assist Trajectories," *Journal of Spacecraft and Rockets*, Vol. 41, No. 5, 2004, pp. 787, 796.

[13] M. V. D. M. Novak, "Improved Shaping Approach to the Preliminary Design of Low-Thrust Trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 1, 2011, pp. 128, 147.

[14] E. A. Ehsan Taheri, Ilya Kolmanovsky, "Shaping low-thrust trajectories with thrust-handling feature," *Advance in Space Research*, 2017, pp. 879, 890.

[15] P. J. Cefola, "Equinoctial Orbital Elements - Application To Artificial Satellite Orbits," *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Palo Alto, California, AIAA, 1972.

[16] H. Curtis, ed., *Orbital Mechanics for Engineering Students*. Oxford: Elsevier, 2005.

[17] V. A. Chobotov, ed., *Orbital Mechanics*. Reston: AIAA Education Series.

[18] O. George P. Sutton, ed., *Rocket Propulsion Elements*, ch. 17. Hoboken: Wiley, 2010.

[19] P. G. A. Quarteroni, F. Saleri, ed., *Calcolo Scientifico*. Springer, 2012.

[20] J. R. W. Wiley J. Larson, ed., *Space Mission Analysis and Design*, p. 407 427. London: Kluver Academy Publisher, 1999.

[21] J. R. F. Michael D. Griffin, ed., *Space Vehicle Design*. Reston: AIAA education Series, 2004.

[22] JPL, ed., *https://dawn.jpl.nasa.gov*. NASA, Oct. 2017.

[23] Arianespace, ed., *VEGA user's Manual*. Arianespace, 2014.

[24] S. J. W. J. Nocedal, ed., *Numerical Optimization*. New York: Springer, 2006.

[25] K. F. Graham and A. V. Rao, "Minimum-Time Trajectory Optimization of Multiple Revolution Low-Thrust Earth-Orbit Transfers," *Journal of Spacecraft and Rockets*, Vol. 52, No. 3, 2015, pp. 711, 727.

[26] S. C. N.R. Ramos, F. Topputo, "Indirect Optimization of Electric Propulsion Orbit Raising to GEO with Homotopy," *Master of Science thesis*, Milan, Italy, Politecnico di Milano, 2017.