J. Prinetto, M. Lavagna
Elliptical Shape-Based Model for Multi-Revolution Planeto-Centric Mission Scenarios
Celestial Mechanics and Dynamical Astronomy, Vol. 133, N. 1, 3, 2021, 3 (24 pages)
doi:10.1007/s10569-020-10001-9

Elliptical shape-based model for multi-revolution planeto-centric mission scenarios

Jacopo Prinetto · **Michelle Lavagna**

Received: date / Accepted: date

Abstract The paper presents a novel 3-dimensional shape-based algorithm which extends the domain of analytical solutions to planeto-centric mission scenarios, which classically entail even thousands revolutions to transfer to the final orbit. Thanks to the strong physical meaning the proposed method keeps while shaping the trajectory, the method succeeds in outputting a solution close to the real optimum. The proposed approach allows to easily formalize practical mission constraints, such as maximum thrust threshold and eclipses; free and fixed time of flight are manageable as well. The approach is almost completely analytic, which is beneficial as it significantly decreases the computational load. It is well suited for complex mission scenarios and for fast detection near optimal solutions to support the whole mission design.

Keywords Low thrust · Trajectory design · Trajectory optimization

J. Prinetto
PhD.Candidate
Politecnico di Milano - Bovisa Campus
Aerospace Science & Technology Dept.
Via La Masa 34, 20156 Milano
Building B12
Tel.: +39-02-2399-8024
E-mail: jacopo.prinetto@polimi.it

M. Lavagna
Full Professor in Flight Mechanics
Politecnico di Milano - Bovisa Campus
Aerospace Science & Technology Dept.
Via La Masa 34, 20156 Milano
Building B12
Tel.: +39 02 2399 8364
E-mail: michelle.lavagna@polimi.it

1 Introduction

In the last decades, Solar Electric Propulsion (SEP) has became of primary interest both for long and complex interplanetary missions (e.g.ESA BepiColombo, NASA Dawn) and for Earth-centered satellite station keeping (e.g. ESA-Artemis, AlphaSat). The continuous progress in SEP related technologies and the flexibility they would offer the mission can extend the applicable domain of these thrusters to the main propulsion system of planeto-centric missions as well, such as tug vehicles and launchers' upper stages. Still, the low thrust trajectory design and optimization represent a challenge. The low thrust trajectory optimisation entails solving quite a complex continuous optimal control problem [47], as the dynamics are formalised by a set of seven coupled non-linear differential equations: in turn, they are built up by the evolution of six state-vector components plus mass rate consumption. The modeling problem grows further in complexity whenever perturbations must be considered and realistic events are to be taken into account (e.g.vehicle separation, gravity assists, eclipses). Perturbations are time and state dependent, while realistic events are formalised as discontinuities. Analytical solutions exist but for simplified, and therefore unrealistic, cases [24] [50]. It is worth noting also that the structure of the solution is typically not a-priori known [26] [47], which makes the powerful direct and indirect optimization methods inapplicable. Moreover, they typically require a good first guess to properly converge as well as having a high computational burden that makes them unsuitable for large search spaces [47][20][27]. Shape-based algorithms aim at coping with the aforementioned limitations. They approach the core strategy and revert the continuous optimal control problem by imposing kinematic constraints to the dynamics: the trajectory shape is selected, with proper degrees of freedom exploited to tune the dynamics compliance. Typically some assumptions, such as the tangential direction of the thrust, are assumed to obtain faster analytical solutions[28][31]. These methods, working on a subspace of the problem, are only capable of giving a sub-optimal solution and are extremely fast if compared with others [33][26][28]. They are well suited for fast detection of sub-optimal solutions in wide search domains using heuristic algorithms, with the possibility to exploit multiple/multidisciplinary objectives [26]. These solutions can either be adopted as initial guesses for direct or indirect optimizations or to obtain independent results during the preliminary phases of a space mission design [26][33]. The first developed shape-based techniques could only solve simple planar problems without the possibility of imposing neither exact boundary conditions on positions and velocities or the time of flight [33]. These algorithms were only used to obtain a quick estimation of the low-thrust trajectory cost and to generate initial trajectories. Improvement in the flexibility of the trajectory and in the precision of the solution was proposed in [30]; the possibility to exactly impose the boundary conditions also provides the opportunity to include gravity assist maneuvers[32]. Conway and Wall developed a simple but effective shape for both 2D problems and approximated 3D problems with small displacements from the plane[31][28]. Recently, Xie et al. [4] and Zeng et al. [8] proposed full 3D shapes for interplanetary trajectory design. Novak and Vasile presented a new coupling between the analytic solution and a LQR controller [21]. Some authors apply Fourier series to tune more effective shapes in finding solutions closer to the optimum [14][7][5].

Gondelach and Noomen [3] developed a shape-based algorithm based on velocity shaping instead of the trajectory that showed good results in interplanetary low thrust trajectory definition. All the above-mentioned shapes, and some other variations proposed by other authors, give very good results in interplanetary trajectory design: the flexibility of the approaches and the reduced computational burden allow designing and optimizing complex mission scenarios, with mixed integer and continuous variables, therefore settling multidisciplinary objectives [26]. Algorithms developed by Taheri and Abdelkhalik [7][6] can efficiently solve Earth-Centered mission scenarios, including rendezvous and phasing, for spacecraft with thrust acceleration in the order of $10^{-1} m/s^2$. Indeed, in this peculiar environment the proximity of the attractor makes the dynamics much more constrained: hundreds or even thousands of revolutions are typically needed to move the satellite between two different orbits, and during the single revolution the osculating elements remain almost unchanged. Purely geometrical shapes cannot fit this behavior, especially whenever eccentric orbits are considered [31][28][33]. Moreover, eclipses introduce a high number of discontinuities, definitely unmanageable with the algorithms mentioned so far. This paper proposes a novel shape-based algorithm applicable to Earth-centered trajectory optimization problems. The working principle of the algorithm is to map the complex multi-revolution problem into a high number of simpler single revolution trajectories, efficiently solved thanks to a new 3D shape-based method. To this end, a proper number of intermediate orbits is introduced and accurately located in order to not to exceed a threshold settled on the required thrust. The rest of the paper is structured as follows: in section 2 the developed single revolution algorithm is discussed; in section 3 the multi-revolution planeto-centric and interplanetary extensions are presented, while section 4 is devoted to the analysis of some test cases.

2 Single Revolution Algorithm

The section presents the developed single-revolution shape-based algorithm. The mathematical formulation is derived in both fixed and free Time Of Flight (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 lies in setting the spacecraft-to-attractor distance and the declination above the reference plane of the trajectory (respectively s and δ) by applying a non-linear interpolation between their values at the departure and target orbits. The kinematic and the dynamics are then recovered through a semi-analytic procedure. The discussion is organized so that, after analysing the requirements, the equation of motion is parametrized, the non-linear interpolation is introduced, and the architecture of the algorithm is explained. Some kinematic relations, even if important in the numerical implementation, are not fundamental for the comprehension of the work. For sake of simplicity these equations are reported only in the appendix.

2.1 Requirements

To drive the design of an algorithm which handles highly constrained dynamics, a peculiarity of the planeto-centric orbits whenever low thrust is selected, some requirements have been set. The most relevant are as follows:

1. The algorithm shall link two different states (Modified Equinoctial Elements [16] or Keplerian Parameters) with the possibility to impose the time of flight.
2. The algorithm shall work with any couple of orbits that can be physically connected with zero radial thrust [28], including polar and retrograde orbits.
3. The thrust acceleration shall be exactly computed via an analytical procedure.
4. The thrust and mass profiles shall be computed via numerical integration of Tsiolkovsky equation [34].
5. The required thrust shall tend to zero as the distance between initial and final orbits decreases, regardless of the eccentricity and departure or arrival anomaly:

$$\lim_{\Delta KP \to 0} \max \left(\frac{|T|}{m} \right) = 0 \qquad \forall \;\; e, i, \theta_1, \theta_2. \tag{1}$$

The last requirement is mandatory to obtain feasible solutions in planeto-centered mission scenarios: whenever not satisfied, the maximum thrust cannot be arbitrarily fixed .

2.2 Low thrust equations parametrization

This section formalizes the parametrization of the 3D Equation of Motion (EoM) for a thrusting spacecraft in cylindrical coordinates. The assumption on small displacements for the out of plane motion, imposed by Wall in [28], is here removed. The exact EoM for a thrusting spacecraft [47] [36] can be written as

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

The working principle that lies behind the proposed algorithm is to parametrize every quantity involved in Eq. 2 as a function of the non-dimensional anomaly x, defined as

$$x(t) = \frac{\theta(t)}{\psi}, \tag{3}$$

and to then compute the thrust and mass time history, adopting a reverse dynamic approach [31]. Physically, $x(t)$ is the angle between the initial position vector and the projection of the spacecraft position vector on the reference plane, normalized by the total transfer angle ψ, as reported in Figure 1.

The spacecraft 3D motion is defined by the parametrization of the in-plane projection of the radius r and the out-of-plane displacement z, as in

$$\begin{cases} r = r(x) \\ z = z(x). \end{cases} \tag{4}$$

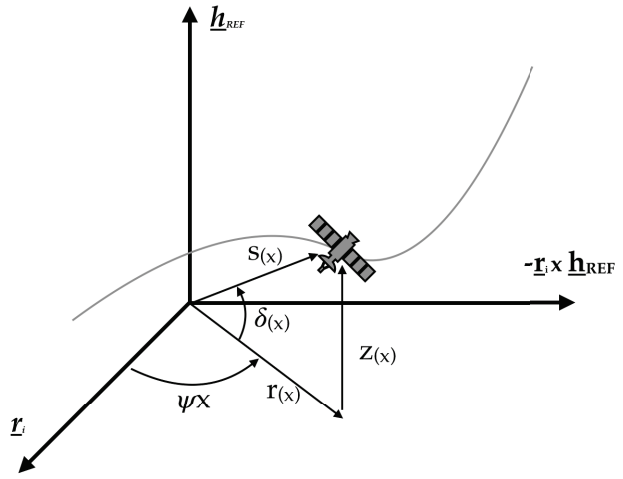

Fig. 1 Spacecraft position

This parametrization, which is not unique, is presented later on in this section. To parametrize the EoM the first and second-time derivatives for the in-plane angular displacement, the in-plane radius and the out-of-plane displacement are first computed, as reported in

$$\begin{cases} \dot{\theta} = \frac{d\theta}{dt} = \psi \dot{x} \\ \ddot{\theta} = \frac{d^2\theta}{dt^2} = \psi \ddot{x} \end{cases} \tag{5}$$

and

$$\begin{cases} \dot{r} = \frac{dr}{dt} = \frac{dr}{dx}\frac{dx}{dt} = r'\dot{x} \\ \ddot{r} = \frac{d^2r}{dt^2} = \frac{d}{dt}\left(r'\dot{x}\right) = r''\dot{x}^2 + r'\ddot{x} \\ \dot{z} = \frac{dz}{dt} = \frac{dz}{dx}\frac{dx}{dt} = z'\dot{x} \\ \ddot{z} = \frac{d^2z}{dt^2} = \frac{d}{dt}\left(z'\dot{x}\right) = z''\dot{x}^2 + z'\ddot{x}. \end{cases} \tag{6}$$

Then, all kinematic quantities can be substituted into Eq. 2, giving as result

$$\begin{cases} r''\dot{x}^2 + r'\ddot{x} - r\psi^2\dot{x}^2 = -\frac{\mu}{s^3}r + \frac{T_{IN}}{m}sin\alpha \\ 2\psi r'\dot{x}^2 + r\psi\ddot{x} = \frac{T_{IN}}{m}cos\alpha \\ z''\dot{x}^2 + z'\ddot{x} = -\frac{\mu}{s^3}z + \frac{T_{OUT}}{m}. \end{cases} \tag{7}$$

According to the first two equations listed in Eq. 7, the formulation for the second time derivative of the non-dimensional anomaly is extracted as follows

$$\begin{cases} \ddot{x} = \frac{1}{r'}\left[-\frac{\mu}{s^3}r + \frac{T_{IN}}{m}sin\alpha - r''\dot{x}^2 + r\psi^2\dot{x}^2\right] \\ \ddot{x} = \frac{1}{r\psi}\left[-2\psi r'\dot{x}^2 + \frac{T_{IN}}{m}cos\alpha\right]. \end{cases} \tag{8}$$

To analytically compute the x time derivative, the dependency from the thrust per unit mass in Eq. 8 must be removed: to this end, the in-plane thrust should be imposed

as tangential only [31]. Indeed, in this last case, the thrust angle equals the flight path angle that can be easily computed using

$$\tan\alpha = \tan\gamma = \frac{v_r}{v_\theta} = \frac{\dot{r}}{r\dot{\theta}} = \frac{r'}{r\psi}. \tag{9}$$

By merging Eq. 9 and Eq. 8 and removing the thrust dependence, the square of x time derivative is made explicit, as follows

$$\dot{x}^2 = \frac{\mu r}{s^3\left(r\psi^2 - r'' + 2\frac{r'^2}{r}\right)} = \frac{Nu}{De}. \tag{10}$$

From Eq. 10, the second time derivative for the non-dimensional anomaly can be obtained as

$$\begin{cases} \ddot{x} = \frac{1}{2}\left(\frac{Nu' - \dot{x}^2 De'}{De}\right) \\ Nu' = \mu r' \\ De' = 3\frac{s'}{s}De + s^3\left(r'\psi^2 - r''' + \frac{2rr'r'' - r'^3}{r^2}\right). \end{cases} \tag{11}$$

By inserting the newly derived time-dependent quantity relationships into Eq. 7, the thrust per unit mass can be computed as

$$\begin{cases} \frac{T_{IN}}{m} = \frac{1}{\cos\gamma}\left(2\psi r'\dot{x}^2 + r\psi\ddot{x}\right) \\ \frac{T_{OUT}}{m} = z''\dot{x}^2 + z'\ddot{x} + \frac{\mu}{s^3}z. \end{cases} \tag{12}$$

The mass time history comes from numerically integrating the Tsiolkovsky equation, as [46]

$$\begin{cases} \left|\frac{T}{m}\right| = \sqrt[2]{\left(\frac{T_{IN}}{m}\right)^2 + \left(\frac{T_{OUT}}{m}\right)^2} \\ \frac{dm}{dt} = -\frac{\left|\frac{T}{m}\right|m}{I_s p g_0}. \end{cases} \tag{13}$$

Moreover, as highlighted in Eq. 14, the integral on the variation of the non-dimensional anomaly, which is numerically solved, leads to the time vector [31]

$$t = \int_0^t \mathrm{d}\tau = \int_0^1 \frac{1}{\dot{x}}\,\mathrm{d}x. \tag{14}$$

It is important to underline that, differently from Wall [28], the here proposed parametrization allows representing the exact motion of a thrusting spacecraft, removing any assumption on small displacements from the reference plane assumption.

2.3 Non-Linear Interpolation

The EoM parametrization discussed so far needs a parametric representation of the trajectory in cylindrical coordinates, as in Eq. 4. In the here presented algorithm the parametrization is obtained via a nonlinear interpolation between the departure and arrival orbits. This interpolation is set in spherical coordinates ($s(x)$, $\delta(x)$, ψx, see Figure 2) by means of an interpolating function $\chi(x)$, as shown in

$$\begin{cases} s_{(x)} = \left(s_{2(x)} - s_{1(x)}\right)\chi_{(x)} + s_{1(x)} \\ \delta_{(x)} = \left(\delta_{2(x)} - \delta_{1(x)}\right)\chi_{(x)} + \delta_{1(x)}; \end{cases} \tag{15}$$

it is then mapped into cylindrical coordinates using

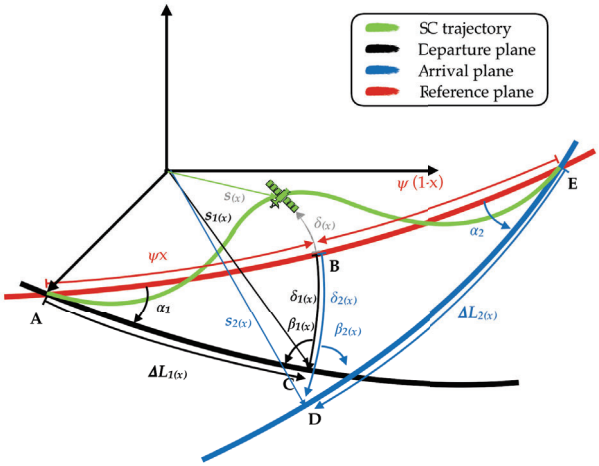

Fig. 2 Non-linear interpolation

$$\begin{cases} r = s\cos\delta \\ z = s\sin\delta. \end{cases} \tag{16}$$

To solve the parametrized equation of motion, the first, second and third derivatives of $r(x)$ and the first two derivatives of $z(x)$ shall be calculated using

$$\begin{cases} r' = -s\delta'\sin\delta + s'\cos\delta \\ r'' = -\left(2s'\delta' + s\delta''\right)\sin\delta + \left(s'' - s\delta'^2\right)\cos\delta \\ r''' = -\left[3\left(s''\delta' + s'\delta''\right) + s\left(\delta''' - \delta'^3\right)\right]\sin\delta + \\ \qquad + \left[s''' - 3\delta'\left(s'\delta' + s\delta''\right)\right]\cos\delta \end{cases} \tag{17}$$

and

$$\begin{cases} z' = s\delta'\cos\delta + s'\sin\delta \\ z'' = \left(2s'\delta' + s\delta''\right)\cos\delta + \left(s'' - s\delta'^2\right)\sin\delta. \end{cases} \tag{18}$$

The computation of

$$\begin{cases} s' = \Delta s' \chi + \chi' \Delta s + s_1' \\ s'' = \Delta s'' \chi + \chi'' \Delta s + 2\Delta s' \chi' + s_1'' \\ s''' = \Delta s''' \chi + \chi''' \Delta s + 3\Delta s'' \chi' + 3\Delta s' \chi'' + s_1''' \end{cases} \tag{19}$$

and

$$\begin{cases} \delta' = \Delta \delta' \chi + \chi' \Delta \delta + \delta_1' \\ \delta'' = \Delta \delta'' \chi + \chi'' \Delta \delta + 2\Delta \delta' \chi' + \delta_1'' \\ \delta''' = \Delta \delta''' \chi + \chi''' \Delta \delta + 3\Delta \delta'' \chi' + 3\Delta \delta' \chi'' + \delta_1''' \end{cases} \tag{20}$$

enables the evaluation of the derivatives of the spherical coordinates included in Eq. 17 and Eq. 18.

The full set of equations needed to compute the geometrical quantities of the initial and final orbits required to solve Eq. 19 and Eq. 20 are reported in Appendix A.

2.4 the algorithm architecture

Figure 3 depicts the implemented algorithm architecture, made up of five main blocks plus two optional, activated whenever requested by the specific mission scenario.

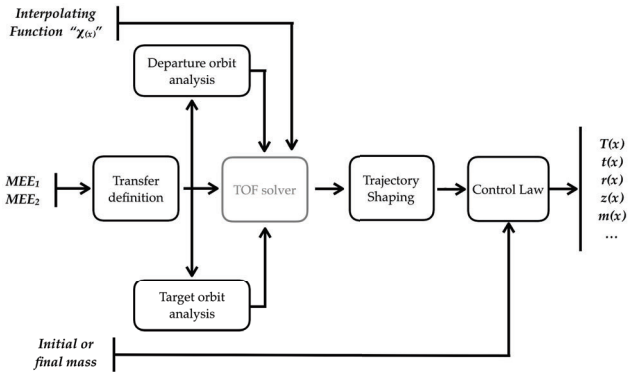

Fig. 3 Architecture

A short description of each block is here reported:

- **Transfer definition:** This block takes as inputs the departure and arrival orbital states; it computes the reference frame and the transfer angle. The selected reference frame is composed by:
  - **First axis:** normalized initial position of the satellite.
  - **Third axis:** normalized cross product between the initial and final positions.
  - **Second axis:** Orthogonal to the first and third axis, to get a right-handed Reference Frame

- **Departure/Target orbit analysis:** These blocks take as input the data coming from the Transfer definition one. They compute some important geometrical quantities exploited for the shape interpolation. The output includes all geometric quantities of the initial and final orbits.
- **TOF solver:** This block manages the the time of flight. It activates only whenever a TOF constrained mission scenario is considered: to this end, the shape itself shall contain a further degree of freedom. To force the imposed time of flight to be respected it is necessary to numerically solve

$$TOF - \int_0^1 \frac{1}{\dot{x}(a)} \mathrm{d}x = 0, \tag{21}$$

  through a Newton algorithm, initialized with $a = 0$.
  At each step of the Newton solver, adopted for integration of Eq. 21, a Cavalieri-Simpson integration scheme is adopted to preserve both a good precision and a limited CPU time. This block outputs the interpolating function, that is passed to the trajectory shaping block.
- **Trajectory shaping:** This block takes as input the geometry of departure and target orbits and the interpolating function. It computes the exact geometry of the transfer using the non-linear interpolation between departure and target orbits explained here-above. The output includes the kinematics of the trajectory.
- **Control Law:** This block takes as input the kinematics of the trajectory. The control law, as well as the time, are computed via a mixed numerical-analytical procedure. To compute the time vector, Eq. 14 is integrated using an high order multi-step predictor-corrector scheme (Adams-Bashford-3 Adams-Multon-4). The same integration scheme is adopted to integrate Tsiolkovsky equation [34] (Eq. 13) to get the mass history. It is here highlighted that both forward and backward integration schemes are supported, making the algorithm suitable to solve scenarios either with dry or wet mass imposed. The block is the most demanding in terms of computational time, as it includes two ODEs to be solved. In any case, it is important to underline that while typically an ODE solver spends most of the time evaluating the function to be integrated, in the framework of the proposed approach those information are directly fed into the solver (please refer to Eq. 14 and Eq. 13), being already computed by previous blocks; that helps in reducing the computational burden of the block itself. Moreover, the selection of an accurate integration scheme reduces the number of computation nodes while keeping the numerical errors as low as possible.

2.5 The interpolating functions

A key element of the developed algorithm lies in the ability of the interpolating function χ to respect the boundary conditions; to this end, the interpolating function $\chi(x)$ must be continuous with its derivatives up to the third order in the domain $[0;1]$. Moreover, the interpolating function must satisfy the following requirements:

– **Position:** from Eq. 16 and Eq. 15 the initial and final conditions on position (in plane radius $r(x)$ and out-of-plane displacement $z(x)$) are automatically satisfied if the interpolating function satisfies:

$$\begin{cases} \chi(0) = 0 \\ \chi(1) = 1. \end{cases} \tag{22}$$

– **Velocity:** given the definition of the radial velocity ($\dot{r}$) and the out-of-plane velocity ($\dot{z}$) in Eq. 6, $r', z', \dot{x}$ 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 transverse 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. 17 and Eq. 18, and are summarized as follows:

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

For the initial and final conditions on $\dot{x}$, Eq. 10 must be considered: it contains the second derivatives of r as well. Therefore, boundary conditions on r'' must be imposed too. From Eq. 17 it is easy to derive the following constraints:

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

Even if an extremely wide number of functions could satisfy those requirements, this work adopts the seventh order polynomial function

$$\chi(x) = -20x^7 + 70x^6 - 84x^5 + 35x^4, \tag{25}$$

and the eight-order polynomial

$$\begin{aligned} \chi(x, a_s) = a_s x^8 - (20 + 4a_s)x^7 + (70 + 6a_s)x^6 + \\ -(84 + 4a_s)x^5 + (35 + a_s)x^4 \end{aligned} \tag{26}$$

to solve the free TOF and the constrained TOF problems respectively.

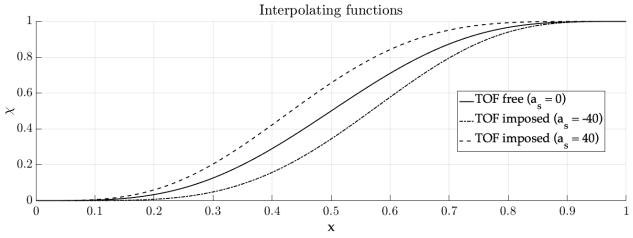

Fig. 4 Interp. function comparison

It is easy to prove that these functions satisfy the above-mentioned boundary conditions. For sake of completeness they are plotted in Figure 4 for different values of the parameter a_s. Moreover, it is here remarked that those interpolation functions lead

to solutions with null initial and final thrust, avoiding the typical undesired peaks at the beginning and the end of the transfer arc (see the thrust profile obtained in [28] and [31]).

The algorithm proposed in the paper, run in MATLAB R2017 on a laptop equipped with a sixth generation Intel i7 processor at 2.6 GHz, can evaluate more than 20 thousand revolution per second for free TOF scenarios and more than 8 thousand for imposed TOF problems.

3 Multi-revolution Approaches

The section discusses two possible solutions to extend the above-mentioned algorithm to the multi-revolutions trajectories. The first solves interplanetary transfers, and it is applicable whenever the number of revolutions is limited (no more than 2 or 3); the second, suitable for planeto-centric mission scenarios, increases in complexity: in fact, it manages thousands of revolutions, including also discontinuities such as eclipses.

3.1 Interplanetary scenario

Due to the low number of complete revolutions that are typically involved in low thrust interplanetary transfers, a simple but effective strategy lies in considering an 'augmented' transfer angle as given in[31][28]

$$\psi = \psi + 2\pi N_{rev}. \tag{27}$$

The solution is well suited for N limited to either 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 trajectory design and optimization introduces some other issues and constraints/objectives to identify and formalize. Regardless of the technology adopted, the thrust is linearly related with the electric power available on board: most of the thrusters on the market show a specific power consumption between 15 $\frac{W}{mN}$ and 40 $\frac{W}{mN}$. During an interplanetary transfer, the distance from Sun varies significantly and, in most cases, it drives the time history of the generated electric power, if photovoltaic technology is adopted on board as the primary power source [48].

Moreover, solar panels are affected by aging effects that reduce the amount of power produced during the mission. These effects can be merged together to formalize a unique constraint that, to speed up the optimizer convergence and its flexibility in the mission design, is included in the cost functions vector and is to be minimized. The solar panel area needed to accomplish the mission is the physical quantity that synthesizes all above-mentioned aspects, which is sized thanks to [48]

$$A_{SA}(x) = \frac{kT + P_{ss}}{\eta_{tot} \cos\phi \, (1-\beta)^t \, \frac{\phi_{Earth}}{s^2}}, \tag{28}$$

at each computational node.

The sun-angle can be either a-priori imposed or computed point-by-point from the control law, if the geometry of the spacecraft is known. Eq. 28 can be simpler as

$$obj = MAX\left[\frac{T(x)s(x)^2}{(1-\beta)^{t(x)}}\right], \tag{29}$$

to generate an objective function less dependent on the spacecraft specific sizing.

This objective function can be successfully included in a multi-objective multidisciplinary heuristic optimization together with the fuel mass minimisation. Moreover, the admissible maximum thrust [26] can be included, formulated either as a constraint or as a further element of the objective functions vector, and is to be minimized. If the latter scheme is adopted, the Pareto front allows directly selecting the solution that fits at the best both the mission requirements and the platform constraints.

3.2 Planeto-centric scenario

The basic strategy proposed in the paper to solve for the planeto-centric scenarios sees the introduction of a set of intermediate keplerian orbits. That scheme opens the possibility to formalize the actual operational case of switching off thrusters during eclipses. The problem is formalized first: the initial and final states are imposed as Modified Equinoctial Elements (MEE):

$$\begin{cases} MEE_i = [p_i, f_i, g_i, h_i, k_i, L_i] \\ MEE_f = [p_f, f_f, g_f, h_f, k_f, L_f]. \end{cases} \tag{30}$$

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

1. **Problem initialization:** at the first algorithmic step the spacecraft is assumed to be in its initial state, while the final state represents the desired condition, therefore

$$\begin{cases} MEE_1 = MEE_i \\ MEE_2 = MEE_f \\ m_k(0) = M_{initial} \\ t_k(0) = 0. \end{cases} \tag{31}$$

   with $k = 1$, as the first trajectory has still to be sized.
2. **k-th intermediate orbit setting:** the spacecraft holds in position described by MEE_1 with mass $m_k(0)$ at time after departure mapped in $t_k(0)$. The goal stays in localizing the k^{th} intermediate orbit such that the required maximum thrust equals the maximum actually available. According to

$$\begin{cases} p_k = (p_2 - p_1)\,\eta_k + p_1 \\ f_k = (f_2 - f_1)\,\eta_k + f_1 \\ g_k = (g_2 - g_1)\,\eta_k + g_1 \\ h_k = (h_2 - h_1)\,\eta_k + h_1 \\ k_k = (k_2 - k_1)\,\eta_k + k_1, \end{cases} \tag{32}$$

the intermediate orbit positioning depends on the η_k parameter: the higher the η_k, the higher the gap between the current and the intermediate orbit, entailing the required thrust to be higher as well.
By numerically solving

$$max\,(T(\eta_k)) - T_{available} = 0, \tag{33}$$

η_k is quantified. The term $max\,(T(\eta_k))$ in Eq. 33 refers to the maximum thrust required during the k^{th} trajectory, computed through the TOF free algorithm presented in 2.
Generally speaking, Eq. 33 numerical solution is not straightforward since continuity and solution existence is not guaranteed. An especially developed hybrid Newton-Bisection algorithm is here adopted: the algorithm tries solving the equation with the Newton method first; whenever that fails, it switches to the Bisection method; if a predefined tolerance cannot be satisfied within a given number of iterations, the equality turns into the following inequality:

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

This last inequality can always be solved since the developed shape-based algorithm fulfills the constraints expressed by

$$\lim_{\Delta MEE \to 0} \max\left(\frac{|T|}{m}\right) = \lim_{\eta_k \to 0} \max\left(\frac{|T|}{m}\right) = 0. \tag{35}$$

A computed η_k either equal or larger than one, means that the available on-board thrust suffices to reach the final position MEE_2, as per Eq. 32: therefore, η_k is automatically switched to one and the k^{th} trajectory is re-computed; the algorithm then stops. A computed η_k between 0 and 1 asks to set an iterative loop. The new starting position is represented by the exit from the eclipse of the k^{th} intermediate orbit, and the initial mass of the spacecraft on the $k+1$ trajectory equals the final mass of the k trajectory, namely:

$$\begin{cases} MEE_1 = MEE_k \\ MEE_2 = MEE_f \\ m_{k+1}(0) = m_k(1) \\ t_{k+1}(0) = t_k(1) + \Delta t_{eclipse}. \end{cases} \tag{36}$$

The time after departure at the beginning of the $k+1$ trajectory equals the arrival time of the k trajectory plus the time spent in shadow ($\Delta t_{eclipse}$).
k is then increased and the algorithm loops up to point 2. The cycle stops as soon as η_k either equals or gets larger than one.

3. **Trajectory analysis**: to preserve computational speed, the previous block limits its outputs to the initial/final mass and the TOF. If more information is needed once the intermediate orbits are placed, the trajectory analysis block is activated to compute all kinematic and dynamic quantities related to the trajectory. The rough order of magnitude for the algorithm computational time is 50%, increased whenever the current block takes place; therefore, while complex mission scenarios optimization holds, it should run on a limited number of solutions.

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

– The initial state is represented by the arrival. The departure state plays the role of the desired conditions. Therefore

$$\begin{cases} MEE_1 = MEE_f \\ MEE_2 = MEE_i \\ m_k(1) = M_{final} \\ t_k(1) = 0, \end{cases} \tag{37}$$

is adopted instead of Eq. 31.
– Using Eq. 37, M_{final} is the final mass, and $t_k(x)$ now means 'time before arrival' instead of 'time after departure' and it is a negative quantity.
– The k^{th} intermediate orbit is integrated itself backward in time, therefore states are updated according to

$$\begin{cases} MEE_1 = MEE_i \\ MEE_2 = MEE_k \\ m_{k+1}(1) = m_k(0) \\ t_{k+1}(1) = t_k(0) - \Delta t_{eclipse}, \end{cases} \tag{38}$$

which is adopted instead of Eq. 36.
The final mass to be imposed to the $k^{th}+1$ single revolution trajectory corresponds to the initial mass computed for the k^{th} revolution, as done to manage the time variable.

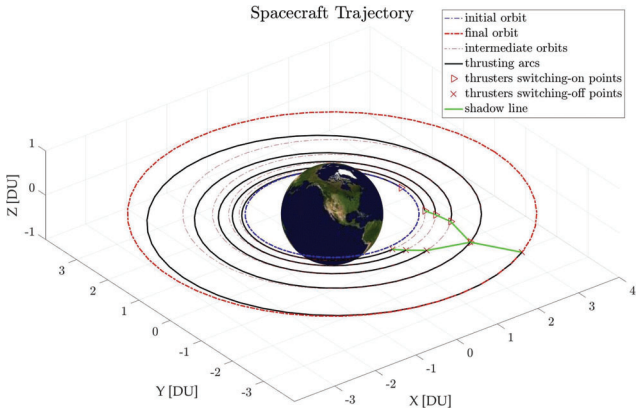

Fig. 5 Example of multi-revolution application

To include both forward and backward directions into the trajectory design, the algorithm broadens applicability to a wider set of scenarios. The algorithm block diagram and flow is offered in Figure 6. Figure 5 sketches how the algorithm works for a multi-revolutionary geocentric trajectory with a high thrust to weight ratio.

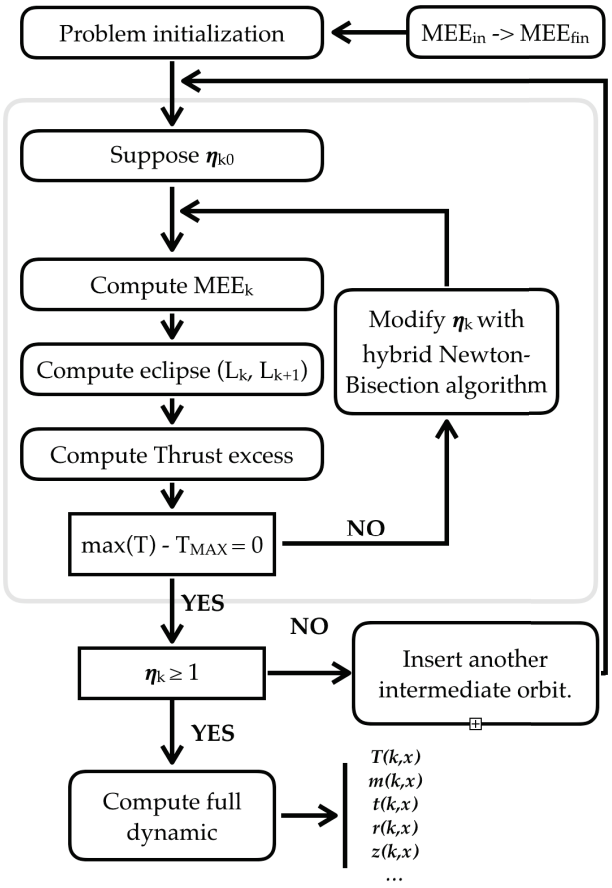

Fig. 6 Multi-revolution algorithm architecture

4 Test Cases

The section discusses some test cases. All runs are accomplished with the MATLAB R2017b coded algorithm on a laptop equipped with a sixth generation Intel i7 processor working at 2.6 GHz with no parallelization.

4.1 Very low thrust LEO-GEO transfer

To perform a comparison between the proposed algorithm verification and literature results, a very low thrust acceleration ($2 \cdot 10^{-4} m/s^2$) planar LEO-GEO transfer scenario has been implemented [2]. Parameters are tuned according to Sreesawet et al. [2], and are reported in Table 1.

The comparison between the solution found by the presented shape-based algorithm and the trajectories sized by Sreesawet et al. [2] is summarized in table 2. The

Table 1 Simulation parameters

Parameter	Value
Initial Mass [kg]	5000
Thrust [N]	1,16
Departure orbit elevation [km]	2000
Specific impulse [s]	1788

Table 2 Comparison of results

	Prinetto et al.		Sreesawet et al.	
Eclipses	on	off	on	off
Final mass [kg]	4026	4025	3980	3993
ToF [days]	267	220	216	202
CPU time [s]	0.37	0.18	15.2	12.5

shape-based algorithm converges to very similar results in terms of fuel consumption (1% better than the solution found by Sreesawet et al.) with a computational time that is two orders of magnitude lower. A gap between 10% and 20% exists on TOF. This undesired gap is mainly due to the limitation imposed by the continuity of the thrust profile.

4.2 Orbital Raising to GEO

The scenario explores the case of raising a satellite to GEO exploiting electric propulsion for energy increase. A satellite dry mass of 800 [kg], equipped with a propulsion unit with 3800 [s] specific impulse and 0.5 [N] thrust, departs from a parking orbit gained by the European VEGA launcher [11], to move to GEO with low thrust. 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 [11]. The apocenter and pericenter radii are degrees 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 Earth rotation axis inclination, 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 [36]. Since in this example the spacecraft is supposed to thrust only in sunlight, the solution depends on the season at GEO arrival too, therefore the two opposite cases (arrival at the equinoxes or at the solstices) have been analyzed. To better highlight the relevant effect the eclipses insertion has on the final solution, comparison is offered with results with the shadowing constraint removed.

Figure 7 depicts the most significant scenario: the GEO injection epoch occurs in the Solstice proximity, and an impulsive disposal maneuver is included in the optimization process to take into account the mandatory reenter of the upper stage 'AVUM' in atmosphere, according to debris mitigation guidelines [1]. Red lines represent the launchable mass with VEGA launcher: since a complete set of information for the launcher is not available, data is extrapolated applying the Tsiolkovsky equation to the launcher upper stage (AVUM) from the reference orbit

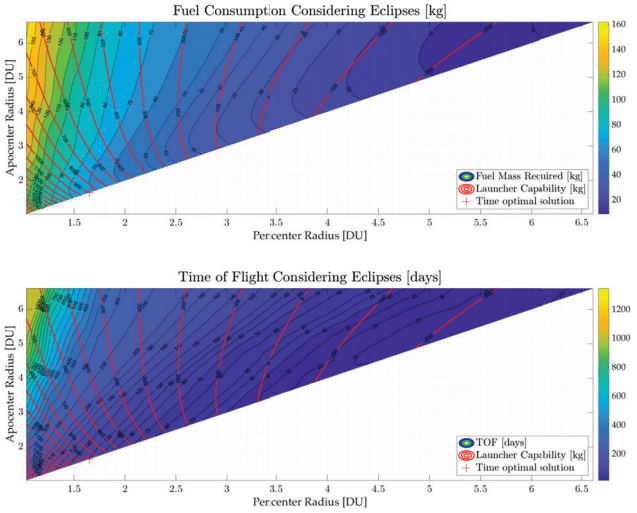

Fig. 7 Orbit rising to GEO

available on the user-manual [11]. The time-optimal problem is solved for all the above-mentioned cases adopting a Nelder-Mead simplex algorithm [35] modified with a penalty method to force the solution detecting an initial mass lower than the launchable mass on the same orbit. Being the number of revolutions discrete, the time of flight is not continuous: therefore the adoption of a derivative free algorithm is mandatory.

Table 3 Time-optimal constrained solutions

Eclipse	Yes		Yes	No
De-orbiting	No	No	Yes	No
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 park. [DU]	1.8994	1.9019	1.6517	1.8816
r_a park. [DU]	1.9030	1.9029	1.6524	1.9207
CPU time [s]	23	31	41	12

Table 3 compares solutions obtained for similar scenarios by slightly changing the framework in terms of arrival epoch, eclipse and disposal maneuver constraints activation. The fuel consumption is similar for scenarios in the table, while the TOF increases between 20% and 25% whenever the no-thrust in shadow constraint is active. Sensitivity to the injection season is contained: the reason why is that, even if nearby equinox eclipses occur at any distance from ground, the fraction of time spent in shadow decreases with the radius. The reported CPU time highlights the algorithm to very quickly find sub-optimal solutions for multi-revolutions discontinuous trajectories: as far as table 3 scenarios are considered, 50 computational nodes per

revolution have been used for the optimization processes and 100 for plotting the final trajectory. The different CPU times reflect the different number of required revolutions to get to GEO, depending on the no-thrust in shadow constraint activation: in fact, it increases the solver complexity and asks for longer thrusting arcs to recover for time lost during the shadowed ballistic arcs. All optimization processes have been initialized with the reference VEGA parking orbit (200 $[km]$ x 1500 $[km]$ height LEO orbits). Literature offers no database with time-optimal solutions of LEO-GEO raising problems including eclipses in the model, therefore bench-marking is feasible with the no-thrust in shadow constraint disabled.

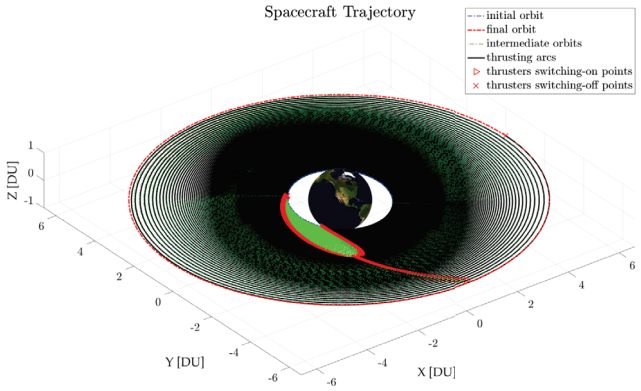

Fig. 8 Time optimal trajectory

While the computed optimal solution is slightly more expensive than those from literature, as will be shown in the section dedicated to the test cases, the CPU time is orders of magnitude lower. Figure 8 shows the attainable 3D trajectory, with a quasi-circular switching orbit with a radius of 2 [DU], with no-thrust in shadow and disposal maneuver constraints active; that indeed, represents the most challenging scenario among those listed in Table 3. It is worth noting the intermediate orbits higher density in Earth proximity. Moreover, the shadow region (in green) is distorted because of the Earth revolution and, less significant, because of the plane changes; the no-thrust region shrinking with altitude, can also be appreciated.

4.3 Earth-Mars Rendezvous

The Earth-Mars rendezvous problem is a classical scenario for the validation of low thrust algorithms [30].

A spacecraft dry mass of 1000 [kg], a specific impulse of 3000 [s] and a maximum thrust of 0.22 [N] are settled. The requested thrust and the fuel mass fraction over the whole search domain are reported in Figure 9; regions where either the thrust exceeds 1 $[N]$ or the fuel mass fraction exceeds 0.5 $[-]$ are white coloured. The optimal thrust and fuel mass fraction are also reported, marked in red: it clearly appears that the search domain proposed by Vasile [21] must be enlarged, being the found optima

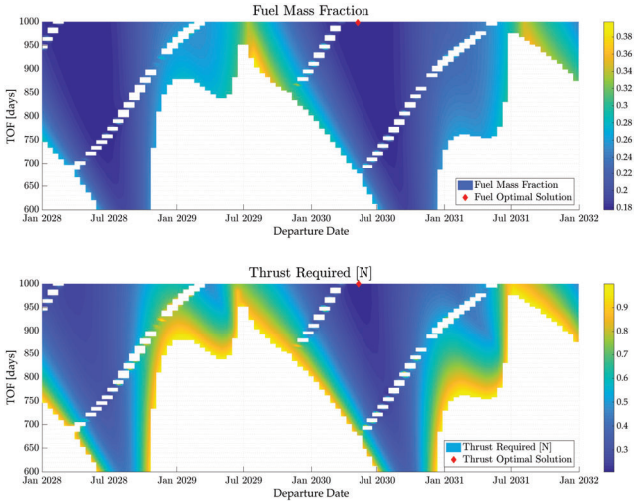

Fig. 9 Launch opportunities

located at its borders. Anyway, staying stuck on that search domain, two convenient regions exist, in which thrust and fuel mass fraction, together with TOF, keep low. The computed fuel optimal solution approaches Vasile and De Pascale results, with the same fuel mass fraction of 0.177 [-] and a 1000 days TOF, slightly higher than the reference; the departure date is the 8^{th} of May 2030. The CPU time, exploiting the standard MATLAB genetic algorithm library with a 100 individuals population for the optimization and a stopping criteria tuned on the average change of the cost function, is lower than 10 seconds.

4.4 Earth-Nereus Rendezvous

The asteroid 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 highly elliptical orbit on a plane slightly above the ecliptic. Its pericenter is located near the Earth's pericenter, while the apocenter is at 1.5 [AU]; therefore, a quasi-ballistic solution with a non-zero escape velocity is expected whenever either the spacecraft fuel mass fraction or the requested thrust level is selected as a cost function. To facilitate the best trajectory finding, an extremely wide search space is considered: the degrees of freedom and their ranges are reported in Table 4.

The optimization is accomplished through the MATLAB genetic algorithm library, with a population of 1000 individuals; the optimal trajectory, according to the stop criterion already introduced, is obtained in 5 minutes with 100 computational nodes.

As can be seen from the output trajectory reported in Figure 10, the launcher directly inserts the spacecraft in a quasi-ballistic orbit, as expected from theory: thrusters are exploited only on arrival at Nereus; slightly more than 10 mN suffice,

Table 4 Search domain

	Range	Optimum
Dep. Date [dd/mm/yyyy]	2030 to 2050	09/02/2042
Time of Flight [days]	500 to 1500	690.5
Number of revolutions [-]	0 to 2	1
v_{inf} departure [km/s]	0 to 6	5.93
v_{inf} in-plane angle [d]	-90 to +90	-8.07
v_{inf} out-plane angle [d]	-90 to +90	44.79

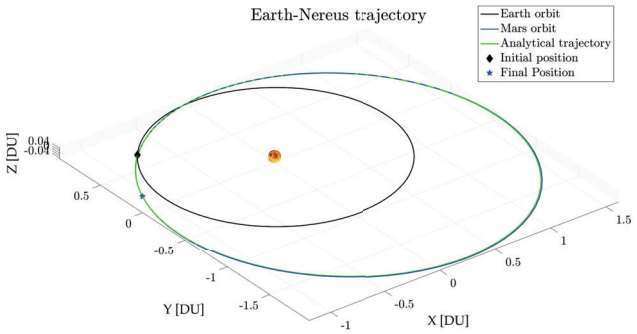

Fig. 10 Earth-Nereus trajectory

and only 0.0052 fuel mass fraction is requested. This example shows that the proposed algorithm can manage also highly elliptical orbits in interplanetary trajectories: that is enabled by the the shape to be a non-linear interpolation between arrival and departure orbits.

5 Final remarks

The paper discusses in detail the proposed shape-based algorithm developed to rapidly generate low-thrust precise trajectories in planeto-centric scenarios. The algorithm shows a good convergence and it is extremely fast and sufficiently flexible to insert operational and platform sizing constraints which affect the trajectory definition. In particular, the implemented strategy to take into account no-thrust in shadow requirements is fast and efficient and works with any class 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 large number of expressions have to be evaluated, they are all composed by the same block of repeated terms that can be computed only once. Moreover, most of the trigonometrical expressions are not directly evaluated by MATLAB functions, but exploiting trigonometrical relations. The next step will include perturbations in the intermediates orbits: the effect will immediately mirror into the interpolated shapes and in the representative dynamics, with the beneficial waterfall in the thrust profile, expected to decrease.

6 Notation

a	Semi-Major axis
a_{pert}	Perturbation acceleration
a_s	Semi-Major axis
A_{sa}	Solar panel 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
KP	Keplerian Elements
L	Sixth MEE
m	Mass
N_{rev}	Number of revolutions
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
Δ_L	Variation of true longitude
η_{tot}	Power production efficiency
θ	Anomaly
ϕ	Solar aspect angle
χ	Interpolating function
ψ	Total transfer angle
ω	angular velocity

A List of derivatives

The appendix reports the equations fundamental to the geometrical interpolation of the trajectory. The subscript '1' indicates the departure orbit, while the subscript '2' indicates the arrival orbit.

A.1 Departure orbit

The inclination of the initial orbit with respect to the reference plane can be computed as in Eq. 39.

$$\begin{cases} \cos\alpha_1 = \hat{\mathbf{h}}_1 \cdot \hat{\mathbf{h}}_{REF} \\ \sin\alpha_1 = \xi_1 \sqrt{1-\cos\alpha_1{}^2} \\ \begin{cases} \xi_1 = 1 & if \quad \mathbf{v}_i \cdot \hat{\mathbf{h}}_{REF} > 0 \\ \xi_1 = -1 & if \quad \mathbf{v}_i \cdot \hat{\mathbf{h}}_{REF} < 0 \end{cases} \end{cases} \tag{39}$$

The declination $(\delta(x)_1)$ of the initial orbit over the reference plane can be computed using Eq. 40, while its derivatives can be computed using Eq. 41, Eq. 42 and Eq. 43.

$$\sin\delta_1 = \sin\alpha_1 \frac{\sin(\psi x)}{\sin\beta_1} \tag{40}$$

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

$$\delta_1'' = \frac{-\psi^2 \sin\alpha_1 \sin(\psi x) + \sin\beta_1 \sin\delta_1 \left(\delta_1'^2 + \beta_1'^2\right)}{\cos\delta_1 \sin\beta_1} + \\ - \frac{2\delta_1'\beta_1' \cos\delta_1 \cos\beta_1 + \beta_1'' \sin\delta_1 \cos\beta_1}{\cos\delta_1 \sin\beta_1} \tag{42}$$

$$\delta_1''' = \frac{-\psi^3 \sin\alpha_1 \cos(\psi x) + \sin\beta_1 \sin\delta_1 \left(3\delta_1'\delta_1'' + 3\beta_1'\beta_1''\right)}{\cos\delta_1 \sin\beta_1} + \\ \cos\beta_1 \frac{-\cos\delta_1 \left(3\delta_1'\delta_1'' + 3\beta_1'\beta_1''\right) + \sin\delta_1 \left(3\delta_1'^2\beta_1' + \beta_1'^3 - \beta_1'''\right)}{\cos\delta_1 \sin\beta_1} \\ + \frac{\sin\beta_1 \cos\delta_1 \left(3\beta_1'^2\delta_1' + \delta_1'^3\right)}{\cos\delta_1 \sin\beta_1} \tag{43}$$

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. 44.

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

$\Delta L_1(x)$ is fundamental to compute the Longitude ($6^{th}MEE$) on the initial orbit at each x and, as a consequence, the attractor distance from the attractor on the departure orbit as function of x; it can be computed with Eq. 45 and Eq. 46.

$$\begin{cases} \sin(\Delta L_1) = \frac{1}{\sin\alpha_1} \sin\delta_1 \\ \cos(\Delta L_1) = \cos(\psi x) \cos\delta_1 \end{cases} \tag{45}$$

$$\begin{cases} \Delta L_1' = \frac{\delta_1'}{\sin\alpha_1 \cos(\psi x)} \\ \Delta L_1'' = \frac{\delta_1'' + \psi \sin\alpha_1 \sin(\psi x) \Delta L_1'}{\sin\alpha_1 \cos(\psi x)} \\ \Delta L_1''' = \frac{\delta_1''' + 2\psi \sin\alpha_1 \sin(\psi x) \Delta L_1'' + \psi^2 \sin\alpha_1 \cos(\psi x) \Delta L_1'}{\sin\alpha_1 \cos(\psi x)} \end{cases} \tag{46}$$

A.2 Target orbit

The inclination of the arrival orbit with respect to the reference plane can be computed as in Eq. 47.

$$\begin{cases} \cos\alpha_2 = \hat{\mathbf{h}}_2 \cdot \hat{\mathbf{h}}_{REF} \\ \sin\alpha_2 = \xi_2\sqrt{1-\cos\alpha_2{}^2} \\ \xi_2 = 1 \qquad if \quad \mathbf{v}_f \cdot \hat{\mathbf{h}}_{REF} < 0 \\ \xi_2 = -1 \qquad if \quad \mathbf{v}_f \cdot \hat{\mathbf{h}}_{REF} > 0 \end{cases} \tag{47}$$

The declination ($\delta(x)_2$) of the arrival orbit over the reference plane can be computed using Eq. 48, while its derivatives can be computed using Eq. 49, Eq. 50 and Eq. 51

$$\sin\delta_2 = \sin\alpha_2 \frac{\sin(\psi(1-x))}{\sin\beta_2} \tag{48}$$

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

$$\delta_2'' = \frac{-\psi^2\sin\alpha_2\sin(\psi(1-x)) + \sin\beta_2\sin\delta_2\left(\delta_2'^2+\beta_2'^2\right)}{\cos\delta_2\sin\beta_2} + \\ - \frac{2\delta_2'\beta_2'\cos\delta_2\cos\beta_2 + \beta_2''\sin\delta_2\cos\beta_2}{\cos\delta_2\sin\beta_2} \tag{50}$$

$$\delta_2''' = \frac{\psi^3\sin\alpha_2\cos(\psi(1-x) + \sin\beta_2\sin\delta_2\left(3\delta_2'\delta_2'' + 3\beta_2'\beta_2''\right)}{\cos\delta_2\sin\beta_2} + \\ \cos\beta_2\frac{\cos\delta_2\left(3\delta_2'\delta_2'' + 3\beta_2'\beta_2''\right) + \sin\delta_2\left(3\delta_2'^2\beta_2' + \beta_2'^3 - \beta_2'''\right)}{\cos\delta_2\sin\beta_2} \\ + \frac{\sin\beta_2\cos\delta_2\left(3\beta_2'^2\delta_2' + \delta_2'^3\right)}{\cos\delta_2\sin\beta_2} \tag{51}$$

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. 52.

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

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. 53 and Eq. 54.

$$\begin{cases} \sin(\Delta L_2) = \frac{1}{\sin\alpha_2}\sin\delta_2 \\ \cos(\Delta L_2) = \cos(\psi(1-x))\cos\delta_2 \end{cases} \tag{53}$$

$$\begin{cases} \Delta L_2' = \frac{\delta_2'}{\sin\alpha_2\cos(\psi(1-x))} \\ \Delta L_2'' = \frac{\delta_2'' - \psi\sin\alpha_2\sin(\psi(1-x))\Delta L_2'}{\sin\alpha_2\cos(\psi(1-x))} \\ \Delta L_2''' = \frac{\delta_2''' + 2\psi\sin\alpha_2\sin(\psi(1-x))\Delta L_2'' + \psi^2\sin\alpha_2\cos(\psi(1-x))\Delta L_2'}{\sin\alpha_2\cos(\psi(1-x))} \end{cases} \tag{54}$$

A.3 Attractor distances

To compute the attractor distance the Longitude at each position x on the initial and final orbits using Eq. 55 and Eq. 56 has to be computed first.

$$\begin{cases} l_1(x) = L_1 + \Delta L_1(x) \\ l_1'(x) = \Delta L_1'(x) \end{cases} \tag{55}$$

$$\begin{cases} l_2(x) = L_2 - \Delta L_2(x) \\ l_2'(x) = -\Delta L_2'(x) \end{cases} \tag{56}$$

The attractor distance on the initial and final orbits can be computed using Eq. 57.

$$\begin{cases} s_i(x) = \frac{p_i}{q_i(x)} \\ s_i(x)' = -\frac{p_i q_i'}{q_i^2} \\ s_i(x)'' = 2\frac{p_i q_i'^2}{q_i^3} - \frac{p_i q_i''}{q_i^2} \\ s_i(x)''' = -6\frac{p_i q_i'^3}{q_i^4} + 6\frac{p_i q_i' q_i''}{q_i^3} - \frac{p_i q_i'''}{q_i^2} \end{cases} \tag{57}$$

the q term appears with its derivatives; it can be computed using Eq. 58

$$\begin{cases} q_i(x) = 1 + f_i \cos l_i(x) + g_i \sin l_i(x) \\ q_i(x)' = (-f_i \sin l_i + g_i \cos l_i)\,\Delta L_i' \\ q_i(x)'' = (1-q_i)\Delta L_i'^2 + q_i' \frac{\Delta L_i''}{\Delta L_i'^2} \\ q_i(x)''' = -q_i' \Delta L_i'^2 + 3(1-q_i)\Delta L_i' \Delta L_i'' + q_i' \frac{\Delta L_i'''}{\Delta L_i'} \end{cases} \tag{58}$$

Conflict of Interest

Authors declare that no conflict of interest holds.

References

1. European Space Agency, ESA Space Debris Mitigation Compliance Verification Guidelines, 2015
2. Sreesawet, Suwat and Dutta, Atri, Fast and robust computation of low-thrust orbit-raising trajectories, Journal of Guidance, Control, and Dynamics, Vol. 41, pag. 1888-1905 (2018)
3. Gondelach, D. J. and Noomen, R., Hodographic-shaping method for low-thrust interplanetary trajectory design, Journal of Spacecraft and Rockets, Vol. 52, pag. 728-738 (2015)
4. Xie, Chengqing and Zhang, Gang and Zhang, Yingchun, Shaping approximation for low-thrust trajectories with large out-of-plane motion, Journal of Guidance, Control, and Dynamics, Vol 39, pag. 2776-2785 (2016)
5. Taheri, Ehsan and Abdelkhalik, Ossama, Initial three-dimensional low-thrust trajectory design, Advances in Space Research, Vol. 57, pag. 889-903 (2016)
6. Abdelkhalik, Ossama and Taheri, Ehsan, Shape Based Approximation of Constrained Low-Thrust Space Trajectories using Fourier Series, Journal of Spacecraft and Rockets, Vol. 49, pag. 535-546 (2012)
7. Abdelkhalik, Ossama and Taheri, Ehsan, Approximate on-off low-thrust space trajectories using fourier series, Journal of Spacecraft and Rockets, Vol. 49, pag. 962-965 (2012)
8. Zeng, Kui and Geng, Yunhai and Wu, Baolin, Shape-based analytic safe trajectory design for spacecraft equipped with low-thrust engines, Aerospace Science and Technology, Vol. 62, pag. 87-97 (2017)
9. Herman, A. L.,and Conway, B. A., Direct Optimization Using Collocation Based on High Order Gauss-Lobatto Quadrature Rules, Journal of Guidance, Control, and Dynamics, Vol 19, pag. 592-599 (1996)
10. Leiter H.L., Ch.Altmann, R.Kukies, J.Kuhmann, J.-P. Porst,Marcel Berger and Michael Rath, Evolution of the AIRBUS DS GmbH Radio Frequency Ion Thruster Family, 34th International Electric Propulsion Conference, Hyogo-Kobe, Japan (2015)
11. Arianespace, VEGA user's Manual, 2014

12. Lavagna, M., Bellei, G., MOPSO Technique Assessment to Cope with First Guess Generation for Interplanetary Trajectories Differently Controlled, 58th International Astronautical Congress, Hyderabad, India, (2007)
13. Dankanich J. W. and Woodcock G. R., Electric Propulsion Performance from Geo-transfer to Geosynchronous Orbits, 30th International Electric Propulsion Conference, Florence, Italy (2007)
14. Ehsan Taheri, Ilya Kolmanovsky, Ella Atkins, Shaping low-thrust trajectories with thrust-handling feature, Advance in Space Research, pag. 879-890 (2017)
15. R. G. Jahn, E. Y. Choueiri, Electric Propulsion, Encyclopedia of Physical Science and Technology, Vol. 5, pag. 125-141 (2002)
16. P. J. Cefola, Equinoctial Orbital Elements - Application To Artificial Satellite Orbits, AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Palo Alto, California, 1972
17. A. Axelrod and M. Guelman,D. Mishne, Optimal Control of Interplanetary Trajectories Using Electrical Propulsion with Discrete Thrust Levels, Journal of Guidance, Control, and Dynamics, Vol. 25, pag. 932-939 (2002)
18. Betts, J.T., Survey of Numerical Methods for Trajectory Optimization, Journal of Guidance, Control, and Dynamics, Vol. 21, pag. 193-207 (1998)
19. Graham, K. F.and Rao,A. V., Minimum-Time Trajectory Optimization of Multiple Revolution Low-Thrust Earth-Orbit Transfers, Journal of Spacecraft and Rockets, Vol. 52 (2015)
20. Kluever C., Oleson, R., Direct Approach for Computing Near-Optimal Low-Thrust Earth-Orbit Transfers, Journal of Spacecraft and Rockets, Vo. 35, pag. 509-515 (1998)
21. Novak, D. M., M. Vasile, Improved Shaping Approach to the Preliminary Design of Low-Thrust Trajectories, Journal of Guidance, Control, and Dynamics, Vol. 34, pag. 128-147 (2011)
22. Betts,J. T., Very low-thrust trajectory optimization using a direct SQP method, Journal of Computational and Applied Mathematics, Journal of Computational and Applied Mathematics, Vol. 120, pag. 27-40, Elsevier (2000)
23. Herman, A.L., Spencer, D.B., Optimal, Low-Thrust Earth-Orbit Transfers Using Higher-Order Collocation Methods, Journal of Guidance, Control, and Dynamics, Vol. 25, pag. 40-47 (2002)
24. Tsien, H.S., Take-Off from Satellite Orbit, Journal of the American Rocket Society, Vol. 23, pag. 233-236 (1953)
25. Betts, J.T., Optimal interplanetary orbit transfers by direct transcription, Journal of the Astronautical Sciences, Vol. 42, pag. 247-268 (1994)
26. Prinetto, J. and Lavagna, M., Main Belt Active Asteroids Samples Collection And Return Mission Design, Italian Association of Aeronautics and Astronautics XXIV International Conference, AIDAA, Palermo-Enna, Italy, 18-22 September 2017
27. Topputo, F. and Zhang,C., Survey of Direct Transcription for Low-Thrust Space Trajectory Optimization with Applications, Abstract and Applied Analysis, Vol. 2014, pag. 15 (2014)
28. Wall, B. J., Shape-Based Approximation Method for Low-Thrust Trajectory Optimization, AIAA/AAS Astrodynamics Specialist Conference and Exhibit, AIAA, 2008, Honolulu, Hawaii
29. Enright P. and Conway,B.A., Discrete Approximations to Optimal Trajectories Using Direct Transcription and Nonlinear Programming, Journal of Guidance, Control, and Dynamics, Vol. 15, pag. 994-1002 (1992)
30. De Pascale P., M. Vasile, Preliminary Design of Low-Thrust Multiple Gravity-Assist Trajectories, Journal of Spacecraft and Rockets, Vol. 43, pag. 1065-1076 (2006)
31. Wall, B.J. and Conway, B.A., Shape-Based Approach to Low-Thrust Rendezvous Trajectory Design, Journal of Guidance, Control, and Dynamics, vol. 32, pag. 95-101 (2009)
32. Petropoulos,A. E. and Longuski,J. M., Shape-Based Algorithm for Automated Design of Low-Thrust. Gravity-Assist Trajectories, Journal of Spacecraft and Rockets, vol. 41, pag 787, 796, (2004)
33. Petropoulos, A. E., John A. Sims, 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
34. Curtis,H., Orbital Mechanics for Engineering Students, Elsevier, Oxford (2005)
35. Nocedal J., Stephen J. Wright, Numerical Optimization, Springer, New York (2006)
36. V. A. Chobotov, Orbital Mechanics, AIAA Education Series, Reston (2002)
37. Schaub.H., and Junkins, J.L., Analytical Mechanics of Space Systems, AIAA Education Series, Reston (2002)
38. JPL, https://dawn.jpl.nasa.gov, NASA, Oct. 2017
39. sitael, http://www.sitael.com, Oct. 2017
40. JAXA, http://global.jaxa.jp/projects/sat, 2018

41. ESA, http://sci.esa.int/bepicolombo/, May 2018
42. ESA, https://earth.esa.int/web/eoportal/satellite-missions/, May 2018
43. NASA, https://www.jpl.nasa.gov/missions/deep-space-1-ds1/, May 2018
44. JPL, https://ssd.jpl.nasa.gov, NASA, May 2018
45. Goebel,D.M., Ira Katz, Fundamentals of Electric Propulsion: Ion and Hall Thrusters, Jet Propulsion Laboratory California Institute of Technology (2008)
46. Sutton, G. P., O.Biblaz, Rocket Propulsion Elements, pag. 622-672, Wiley, Hoboken (2010)
47. Conway,B. A., Spacecraft Trajectory Optimization, pag. 37-76, Cambridge University Press, Cambridge (2010)
48. Wiley J. Larson and James R. Wertz, Space Mission Analysis and Design, pag. 407-427, Kluver Academy Publisher, London (1999)
49. Griffin,M., James R. French, Space Vehicle Design, chapter 5, AIAA education Series, Reston (2004)
50. Battin, R.H., An Introduction to the Mathematics and Methods of Astrodynamics, chapter 1, AIAA education Series, Reston (1999)