A.C. Morelli, C. Hofmann, F. Topputo
Robust Low-Thrust Trajectory Optimization Using Convex Programming and a Homotopic Approach
IEEE Transactions on Aerospace and Electronic Systems, In press - Published online 17/11/2021
doi:10.1109/TAES.2021.3128869

Robust Low-Thrust Trajectory Optimization Using Convex Programming and a Homotopic Approach

Andrea Carlo Morelli, Christian Hofmann, Francesco Topputo

***Abstract*—A robust algorithm to solve the low-thrust fuel-optimal trajectory optimization problem for interplanetary spacecraft is developed in this paper. The original nonlinear optimal control problem is convexified and transformed into a parameter optimization problem using an arbitrary-order Gauss–Lobatto discretization scheme with nonlinear control interpolation. A homotopic approach that considers the energy-to-fuel smoothing path is combined with an adaptive second-order trust-region mechanism to increase performance. The overall robustness is assessed in several fuel-optimal transfers with poor initial guesses. The results show a superior performance in terms of convergence and computational time compared to standard convex programming approaches in the literature.**

I. Introduction

In recent years, the miniaturization of satellites and of their components has significantly decreased the costs of space missions [1]. Small satellites such as CubeSats have granted institutions and small companies access to space, resulting in an increasing number of launches. As all spacecraft are operated similarly, the operational costs of CubeSats are comparable to those of conventional spacecraft [2]. Increasing the level of autonomy and shifting flight-related tasks such as the guidance design on-board is therefore a desirable goal for future missions [3]. These developments require new trajectory design approaches.

Designing a low-thrust trajectory is a complex task as it requires solving a nonlinear optimal control problem. State-of-the-art methods can be compared in terms of computational effort (how many computational resources are needed), reliability (capability of converging even when a poor initial guess is provided) and optimality (minimization of some cost function) [4]. Given the high computing power of desktop computers and workstations, computational time and effort are often of secondary importance; the focus is on optimality. Yet, preliminary studies and scenarios like autonomous guidance require computationally fast and robust techniques that are able to find (near) optimal solutions in little time. This is of paramount importance for on-board applications because the algorithm must be compliant with the limited resources and repeatedly compute feasible trajectories in real-time.

In [5], an exhaustive overview of the existing strategies for optimal trajectory generation is given. Classical direct methods that solve the full nonlinear optimization problem are in general not suitable due to their computational complexity and poor robustness [6], [7]. Likewise, indirect methods that make use of the calculus of variations are often not a viable option as they have poor convergence properties [8]. Convex optimization represents instead an interesting direct approach when dealing with real-time applications since convex problems can be solved by means of polynomial-time algorithms and limited computational resources [9]–[11]. In this context, nonconvex problems are usually transformed into a convex program by means of two specific techniques: lossless convexification [12], [13] (also called exact convex relaxation [14]) and successive convexification [15]; they are then solved by means of an iterative technique [16], [17]. This trust-region based sequential convex programming (SCP) algorithm has recently been applied to different space-related optimization problems, including powered descent and landing [18], entry [19], [20], and low-thrust trajectory optimization [21]. In [4], the minimum-fuel low-thrust trajectory optimization problem in spherical coordinates was convexified and solved with a simple trapezoidal discretization method. In [7], an adaptive flipped Radau pseudospectral convex programming method was combined with a bang-off-bang mesh refinement procedure to compute more complex fuel-optimal interplanetary trajectories.

Due to the discontinuous control, however, current methods may encounter problems finding feasible fuel-optimal solutions directly. In indirect optimization, smoothing techniques and homotopic approaches are used to overcome this issue. A sequence of simpler, smooth problems is solved first and a continuation is performed until the original, discontinuous problem is eventually solved. A logarithmic smoothing function and a hyperbolic tangent function were considered in [22] and [23], respectively. In [24], [25], an energy-to-fuel continuation was used to generate minimum-fuel low-thrust trajectories. In [26], a cubic function of the thrust as a homotopic path was used, while in [27], [28] the performance of additional smoothing functions was compared. In [29], a double-homotopy technique was developed to find the minimum-fuel transfer in the circular restricted three-body problem.

This work proposes a homotopic approach to improve the robustness of a convex programming algorithm when applied to the fuel-optimal low-thrust trajectory optimization problem. A refined SCP algorithm based on an arbitrary-order Gauss–Lobatto discretization scheme is used, which exploits a nonlinear interpolation of the control variables. An adaptive second-order trust-region mechanism is developed to increase

A. C. Morelli and C. Hofmann are PhD Candidates at the Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, 20158 Milano, Italy, E-mail: (andreacarlo.morelli@polimi.it, christian.hofmann@polimi.it).

F. Topputo is Full Professor at the Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, 20158 Milano, Italy, E-mail: francesco.topputo@polimi.it.

the convergence rate. The purpose of the paper is to show the effectiveness of our approach in numerical simulations with poor initial guesses.

The paper is structured as follows. Section II states the trajectory optimization problem. Section III describes the arbitrary-order Gauss–Lobatto discretization scheme for convex programming applications. The homotopic approach and the complete SCP algorithm are presented in Section IV. Section V shows the numerical results, and Section VI reports the final remarks.

II. Problem Formulation

The minimum-fuel space trajectory optimization (STO) problem aims to find the trajectory between two given points with minimum propellant expenditure. Considering two-body dynamics only, the equations of motion in Cartesian coordinates are given by

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{r}} \\ \dot{\mathbf{v}} \\ \dot{m} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ -\mu\mathbf{r}/r^3 + \tilde{\mathbf{u}}_\alpha/m \\ -T/(I_{\text{sp}} g_0) \end{bmatrix} \tag{1}$$

where $\mathbf{r} = [r_x, r_y, r_z]^\top$, $\mathbf{v} = [v_x, v_y, v_z]^\top$, and m are the position vector, the velocity vector, and the mass of the spacecraft, respectively. μ is the standard gravitational parameter of the primary body, $\tilde{\mathbf{u}}_\alpha = [T_x, T_y, T_z]^\top$, $T = \|\tilde{\mathbf{u}}_\alpha\|_2$, and $\tilde{\mathbf{u}} = [\tilde{\mathbf{u}}_\alpha, T]^\top$ is the control vector. I_{sp} and g_0 are the specific impulse and gravitational acceleration, respectively. Eq. (1) is nonlinear and nonconvex; moreover, the state and control variables are coupled through the term $\tilde{\mathbf{u}}_\alpha/m$. Therefore, we transform the original minimum-fuel STO problem into a convex one [4], [7]:

$$\min_{\mathbf{u}} J_0 = \int_{t_0}^{t_f} \tau(t)\ \mathrm{d}t \tag{2}$$

subject to:

$$\dot{\mathbf{x}} = \mathbf{f}_f(\mathbf{x}^*) + \mathbf{A}(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) + \mathbf{B}\mathbf{u} \tag{3a}$$

$$\tau_x^2 + \tau_y^2 + \tau_z^2 \leq \tau^2 \tag{3b}$$

$$0 \leq \tau \leq T_{\max} e^{-w^*}[1 - (w - w^*)] \tag{3c}$$

$$\mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u \tag{3d}$$

$$\mathbf{u}_{\alpha,l} \leq \mathbf{u}_\alpha \leq \mathbf{u}_{\alpha,u} \tag{3e}$$

$$\|\mathbf{x} - \mathbf{x}^*\|_1 \leq R \tag{3f}$$

$$\mathbf{x}(t_0) = \mathbf{x}_0, \quad \mathbf{r}(t_f) = \mathbf{r}_f, \quad \mathbf{v}(t_f) = \mathbf{v}_f \tag{3g}$$

$$\tag{3h}$$

with states $\mathbf{x} = [\mathbf{r}, \mathbf{v}, w]^\top$ and controls $\mathbf{u} = [\tau_x, \tau_y, \tau_z, \tau]^\top$. Note that a change of variables has been exploited to replace the mass m and the thrust T such that $w = \ln m$ and $\tau = T/m$ (τ_x, τ_y and τ_z are defined accordingly). Eqs. (2) and (3) represent the convex STO problem, which will be referred to as CXP (Convex Problem) throughout this paper. As it will be extensively explained in Section IV, directly solving CXP does not correspond to solving the original nonconvex minimum-fuel STO problem. An iterative technique that considers a sequence of convex problems is, in fact, needed. The dynamics in (1) have been linearized to obtain (3a), where the superscript $(\cdot)^*$ denotes the reference trajectory. In (3a),

$$\mathbf{f}_f = [v_x, v_y, v_z, -r_x/r^3, -r_y/r^3, -r_z/r^3, 0]^\top \tag{4}$$

denotes the vector of the natural two-body dynamics, $\mathbf{A} = \partial\mathbf{f}_f/\partial\mathbf{x}$ the Jacobian matrix and $\mathbf{B}$ is such that

$$\mathbf{B} = \begin{bmatrix} \mathbf{0}_{3\times 4} \\ \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{1\times 3} & b \end{bmatrix} \tag{5}$$

where $b = -1/(I_{\text{sp}} g_0)$. The relationship between the components of the control vector has been convexified in (3b); the inequality constraint on τ in (3c) has been linearized due to the change of variables [4]. The lower (subscript l) and upper bounds (subscript u) of states and controls are given in (3d) and (3e), respectively. The trust-region constraint in (3f) is used to keep the solution close to the reference and hence, the linearization valid. Initial $\mathbf{x}_0$ and final $(\mathbf{r}_f, \mathbf{v}_f)$ boundary conditions are given in (3g).

III. Convex Arbitrary-Order Legendre–Gauss–Lobatto Quadrature

CXP in (2) and (3) represents an infinite-dimensional optimal control problem. There are several discretization schemes to transform it into a finite-dimensional parameter optimization problem [30]. A popular choice are pseudospectral methods because of their spectral convergence rate for smooth problems [7], [31]. Yet, the discretized problem is often less sparse compared to zero-order-hold or first-order-hold interpolation [32], local methods such as the trapezoidal rule (linear interpolation of states and controls) and higher order methods like the Hermite–Simpson scheme (cubic interpolation of states and linear interpolation of controls). A generalization of Hermite–Simpson is the arbitrary-order Gauss–Lobatto collocation method that is widely used to solve nonlinear programs (NLP) [33]. In this paper, we apply this method to convex optimization and choose the Legendre–Gauss–Lobatto points for both the nodes and the collocation points. In [34], the performance of the zero-order-hold, first-order-hold, Runge-Kutta and of three pseudospectral discretization methods was compared in space-related convex optimization problems. However, to the best of the authors' knowledge, an arbitrary-order Legendre–Gauss–Lobatto method based on Hermite interpolation has never been applied to convex programs instead. Therefore, this work aims to complement the existing literature on discretization methods for convex optimization in space-related applications.

A. State Discretization

The arbitrary-order Gauss–Lobatto collocation method approximates the state variables by means of an arbitrary-order polynomial and is therefore an extension of the Hermite–Simpson scheme. The total time of flight (ToF) is divided into I subintervals. Each time interval $[t_i, t_{i+1}]$ is mapped into the interval $[-1, 1]$ through the transformation

$$t \ \rightarrow\ \frac{h}{2}\xi + \frac{t_{i+1} + t_i}{2} \tag{6}$$

where $\xi \in [-1,1]$ and $h = t_{i+1} - t_i$ is the time step. Inside the i-th subinterval, the state $\mathbf{x}^{(i)}(\xi) \in \mathbb{R}^{n_x \times 1}$ ($n_x = 7$) is approximated as

$$\mathbf{x}^{(i)}(\xi) \approx \mathbf{a}_0^{(i)} + \mathbf{a}_1^{(i)}\xi + \cdots + \mathbf{a}_n^{(i)}\xi^n, \quad i = 1, \ldots, I \tag{7}$$

where the column vectors of coefficients $\mathbf{a}_m^{(i)} \in \mathbb{R}^{n_x \times 1}$, $m = 0, \ldots, n$ are unknowns that are used to approximate the state at the nodal and collocation points. The idea is to find the values of the coefficients $\mathbf{a}_m^{(i)}$ using the information of the states and dynamics at the nodes (the so-called *Hermite interpolation*) and eventually express the constraints at the collocation points. In this paper, nodes and collocation points are defined according to the Legendre–Gauss–Lobatto (LGL) points, which are the roots of the derivative of the $(n-1)$-th degree Legendre polynomial [33]. The n-th degree Legendre polynomial is given by the Rodrigues formula as [35]

$$P_n(\xi) = \frac{(-1)^n}{2^n n!} \frac{\mathrm{d}^n}{\mathrm{d}\xi^n}(1-\xi^2)^n \tag{8}$$

and the LGL points are then

$$\hat{\boldsymbol{\xi}} = [\hat{\xi}_1 = -1, \hat{\xi}_2, \ldots, \hat{\xi}_i, \ldots, \hat{\xi}_{n-1}, \hat{\xi}_n = 1] \tag{9}$$

where $\hat{\xi}_i$ ($i = 2, \ldots, n-1$) is the i-th root of the polynomial $\mathrm{d}P_{n-1}(\xi)/\mathrm{d}\xi$. The nodes θ_j and collocation points ζ_j are defined as [30]

$$\theta_j = \hat{\xi}_{2j-1}, \quad j = 1, \ldots, (n+1)/2 \tag{10}$$

and

$$\zeta_j = \hat{\xi}_{2j}, \quad j = 1, \ldots, (n-1)/2 \tag{11}$$

Note that only odd orders $n \geq 3$ can be considered, since the number of collocation points inside each interval is defined as $(n-1)/2$ [33]. Fig. 1 illustrates the nodes and collocation points according to the Legendre distribution for different Gauss–Lobatto orders. Considering the information at the nodes, the following linear system can be written for the i-th trajectory segment [33], [36]:

$$\underbrace{\begin{bmatrix} \mathbf{I}_{n_x} & \theta_1\mathbf{I}_{n_x} & \ldots & \theta_1^n\mathbf{I}_{n_x} \\ \mathbf{I}_{n_x} & \theta_2\mathbf{I}_{n_x} & \ldots & \theta_2^n\mathbf{I}_{n_x} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{I}_{n_x} & \theta_{n_p}\mathbf{I}_{n_x} & \ldots & \theta_{n_p}^n\mathbf{I}_{n_x} \\ \mathbf{0}_{n_x} & \mathbf{I}_{n_x} & \ldots & n\theta_1^{n-1}\mathbf{I}_{n_x} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_{n_x} & \mathbf{I}_{n_x} & \ldots & n\theta_{n_p}^{n-1}\mathbf{I}_{n_x} \end{bmatrix}}_{\boldsymbol{\theta}} \underbrace{\begin{bmatrix} \mathbf{a}_0^{(i)} \\ \vdots \\ \mathbf{a}_{n_p}^{(i)} \\ \vdots \\ \mathbf{a}_{n-1}^{(i)} \\ \mathbf{a}_n^{(i)} \end{bmatrix}}_{\mathbf{a}^{(i)}} = \underbrace{\begin{bmatrix} \mathbf{x}^{(i)}(\theta_1) \\ \vdots \\ \mathbf{x}^{(i)}(\theta_{n_p}) \\ \frac{h}{2}\mathbf{f}_l^{(i)}(\theta_1) \\ \vdots \\ \frac{h}{2}\mathbf{f}_l^{(i)}(\theta_{n_p}) \end{bmatrix}}_{\mathbf{b}^{(i)}} \tag{12}$$

where θ_j are the positions of the nodal points as in (10), $n_p = (n+1)/2$ is the number of nodes in each time interval, $\mathbf{I}_{n_x}$ is the $n_x \times n_x$ identity matrix and $\mathbf{0}_{n_x}$ the $n_x \times n_x$ null matrix. Note that $\mathbf{f}_l(\theta_j)$ is the two-body dynamics as given in (3a) and thus

$$\mathbf{f}_l(\theta_j) = \mathbf{f}_f(\mathbf{x}_j^*) + \mathbf{A}(\mathbf{x}_j^*)(\mathbf{x}_j - \mathbf{x}_j^*) + \mathbf{B}\mathbf{u}_j \tag{13}$$

where the subscript $(\cdot)_j$ indicates that the considered quantity is evaluated at the node θ_j and the superscript $(\cdot)^{(i)}$ has been dropped for simplicity. This linearization is required due to our convex approach. The linear system in Eq. (12) can be written in compact form as $\boldsymbol{\theta}\mathbf{a}^{(i)} = \mathbf{b}^{(i)}$ and hence $\mathbf{a}^{(i)} = \boldsymbol{\theta}^{-1}\mathbf{b}^{(i)}$, with $\boldsymbol{\theta} \in \mathbb{R}^{2n_x n_p \times 2n_x n_p}$, $\mathbf{a}^{(i)} \in \mathbb{R}^{2n_x n_p \times 1}$ and $\mathbf{b}^{(i)} \in \mathbb{R}^{2n_x n_p \times 1}$. The state at the collocation points is defined as

$$\mathbf{x}^{(i)}(\boldsymbol{\zeta}) = [\mathbf{x}^{(i)}(\zeta_1); \ldots; \mathbf{x}^{(i)}(\zeta_j); \ldots; \mathbf{x}^{(i)}(\zeta_{n_c})] \in \mathbb{R}^{n_c n \times 1} \tag{14}$$

where $n_c = (n-1)/2$ is the number of collocation points in one interval. Considering (7) and recalling that $\mathbf{a}^{(i)} = \boldsymbol{\theta}^{-1}\mathbf{b}^{(i)}$, the state can be calculated as

$$\begin{aligned} \mathbf{x}^{(i)}(\boldsymbol{\zeta}) &= \underbrace{\begin{bmatrix} \mathbf{I}_{n_x} & \zeta_1\mathbf{I}_{n_x} & \ldots & \zeta_1^n\mathbf{I}_{n_x} \\ \mathbf{I}_{n_x} & \zeta_2\mathbf{I}_{n_x} & \ldots & \zeta_2^n\mathbf{I}_{n_x} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{I}_{n_x} & \zeta_{n_c}\mathbf{I}_{n_x} & \ldots & \zeta_{n_c}^n\mathbf{I}_{n_x} \end{bmatrix}}_{\boldsymbol{\zeta}} \underbrace{\begin{bmatrix} \mathbf{a}_0^{(i)} \\ \mathbf{a}_1^{(i)} \\ \vdots \\ \mathbf{a}_n^{(i)} \end{bmatrix}}_{\mathbf{a}^{(i)}} \\ &= \boldsymbol{\zeta}\boldsymbol{\theta}^{-1}\mathbf{b}^{(i)} = \boldsymbol{\phi}\mathbf{b}^{(i)} \end{aligned} \tag{15}$$

where $\boldsymbol{\zeta} \in \mathbb{R}^{n_c n_x \times 2n_x n_p}$ and $\boldsymbol{\phi} \equiv \boldsymbol{\zeta}\boldsymbol{\theta}^{-1}$. The derivative of the state at the collocation points can therefore be written as

$$\begin{aligned} \frac{\mathrm{d}\mathbf{x}^{(i)}(\boldsymbol{\zeta})}{\mathrm{d}\xi} &= \underbrace{\begin{bmatrix} \mathbf{0}_{n_x} & \mathbf{I}_{n_x} & \ldots & n\zeta_1^{n-1}\mathbf{I}_{n_x} \\ \mathbf{0}_{n_x} & \mathbf{I}_{n_x} & \ldots & n\zeta_2^{n-1}\mathbf{I}_{n_x} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_{n_x} & \mathbf{I}_{n_x} & \ldots & n\zeta_{n_c}^{n-1}\mathbf{I}_{n_x} \end{bmatrix}}_{\boldsymbol{\zeta}'} \underbrace{\begin{bmatrix} \mathbf{a}_0^{(i)} \\ \mathbf{a}_1^{(i)} \\ \vdots \\ \mathbf{a}_n^{(i)} \end{bmatrix}}_{\mathbf{a}^{(i)}} \\ &= \boldsymbol{\zeta}'\boldsymbol{\theta}^{-1}\mathbf{b}^{(i)} = \boldsymbol{\phi}'\mathbf{b}^{(i)} \end{aligned} \tag{16}$$

Note that $\mathrm{d}\mathbf{x}^{(i)}(\boldsymbol{\zeta})/\mathrm{d}\xi$ in Eq. (16) indicates the derivative of the state with respect to the independent variable ξ at the collocation points $\boldsymbol{\zeta}$, where we used the same notation as in [33]. These expressions can easily be extended to all trajectory segments by defining the vectors of unknown coefficients $\hat{\mathbf{a}}$ and constant terms $\hat{\mathbf{b}}$ as

$$\begin{aligned} \hat{\mathbf{a}} &= [\mathbf{a}^{(1)}; \ldots; \mathbf{a}^{(i)}; \ldots; \mathbf{a}^{(I)}] \\ \hat{\mathbf{b}} &= [\mathbf{b}^{(1)}; \ldots; \mathbf{b}^{(i)}; \ldots; \mathbf{b}^{(I)}] \end{aligned} \tag{17}$$

Note that the first node of the i-th subsegment and the last node of the $(i-1)$-th subsegment are equal. Consequently, the vectors $\mathbf{x}^{(i)}(\theta_1)$ and $\mathbf{f}_l^{(i)}(\theta_1)$ in $\mathbf{b}^{(i)}$ are the same as the vectors $\mathbf{x}^{(i-1)}(\theta_{n_p})$ and $\mathbf{f}_l^{(i-1)}(\theta_{n_p})$ in $\mathbf{b}^{(i-1)}$ for $i = 2, \ldots, I-1$. Extending the linear system in (12) to all trajectory segments yields

$$\boldsymbol{\Theta}\hat{\mathbf{a}} = \hat{\mathbf{b}} \iff \hat{\mathbf{a}} = \boldsymbol{\Theta}^{-1}\hat{\mathbf{b}} \tag{18}$$

where $\boldsymbol{\Theta} \in \mathbb{R}^{2In_x n_p \times 2In_x n_p}$ is a diagonal matrix with $\boldsymbol{\theta}$ on the main diagonal. Defining $\hat{\mathbf{x}}(\boldsymbol{\zeta}) = [\mathbf{x}^{(1)}(\boldsymbol{\zeta}); \mathbf{x}^{(2)}(\boldsymbol{\zeta}); \ldots; \mathbf{x}^{(i)}(\boldsymbol{\zeta}); \ldots; \mathbf{x}^{(I)}(\boldsymbol{\zeta})]$ as the column vector of concatenated states at the collocation points, we obtain

$$\hat{\mathbf{x}}(\boldsymbol{\zeta}) = \underbrace{\mathbf{Z}\boldsymbol{\Theta}^{-1}}_{\boldsymbol{\Phi}}\hat{\mathbf{b}} \equiv \boldsymbol{\Phi}\hat{\mathbf{b}} \tag{19}$$

$$\frac{\mathrm{d}\hat{\mathbf{x}}(\boldsymbol{\zeta})}{\mathrm{d}\xi} = \underbrace{\mathbf{Z}'\boldsymbol{\Theta}^{-1}}_{\boldsymbol{\Phi}'}\hat{\mathbf{b}} \equiv \boldsymbol{\Phi}'\hat{\mathbf{b}} \tag{20}$$

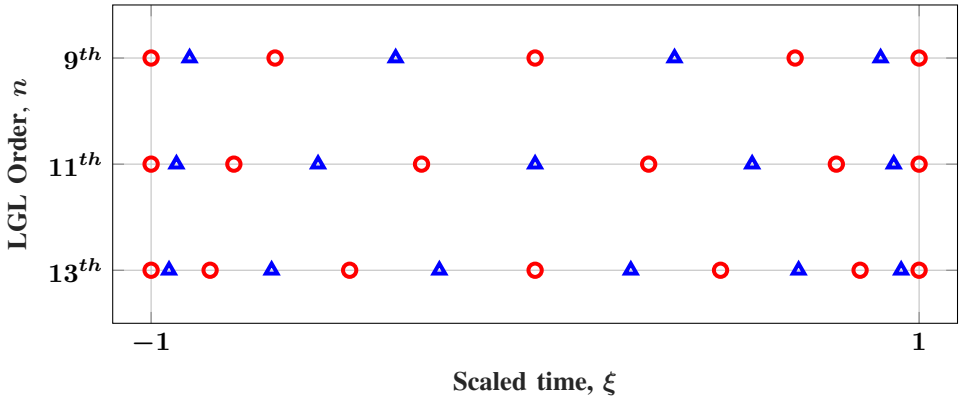


Fig. 1: Nodes θ_j (circles) and collocation points ζ_j (triangles) for different Legendre–Gauss–Lobatto orders.

$\mathbf{Z}$ and $\mathbf{Z}'$ are diagonal matrices with $\boldsymbol{\zeta}$ and $\boldsymbol{\zeta}'$ on the main diagonal, respectively. These expressions are now used to write the defect constraints $\boldsymbol{\Delta}$ in a compact way:

$$\boldsymbol{\Delta} = \boldsymbol{\Phi}'\hat{\mathbf{b}} - \frac{h}{2}[\hat{\mathbf{f}}_f + \hat{\mathbf{A}}(\boldsymbol{\Phi}\hat{\mathbf{b}} - \boldsymbol{\Phi}\hat{\mathbf{b}}^*)] + \hat{\mathbf{B}}\hat{\mathbf{u}}(\zeta) = \mathbf{0} \tag{21}$$

with

$$\hat{\mathbf{f}}_f = [\mathbf{f}_{f,1}^{(1)}, \ldots, \mathbf{f}_{f,n_c}^{(1)}, \ldots, \mathbf{f}_{f,1}^{(I)}, \ldots, \mathbf{f}_{f,n_c}^{(I)}]^\top \tag{22}$$

$$\begin{aligned} \hat{\mathbf{A}} &= \begin{bmatrix} \mathbf{A}(\zeta_1) & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}(\zeta_2) & & \vdots \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \ldots & \ldots & \mathbf{A}(\zeta_p) \end{bmatrix} \\ \hat{\mathbf{B}} &= \begin{bmatrix} \mathbf{B} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{B} & & \vdots \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \ldots & \ldots & \mathbf{B} \end{bmatrix} \end{aligned} \tag{23}$$

$p = In_c$ is the total number of collocation points and $\mathbf{A}(\zeta_j) = \mathbf{A}(\mathbf{x}(\zeta_j))$. The concatenated control vector $\hat{\mathbf{u}}(\zeta)$ will be described in the next section.

Remark 1: Different from the conventional LGL scheme, we include the initial and final points in the optimization to easily account for the initial and final boundary conditions.

B. Nonlinear Control Interpolation

Unlike the state, the control variables are often linearly interpolated in direct collocation methods [30]. As this may result in a poor approximation, we approximate the controls $\mathbf{u}^{(i)}(\xi) \in \mathbb{R}^{n_u \times 1}$ (n_u = 4 is the number of control variables) in the i-th subinterval as a polynomial of degree $n_p - 1$:

$$\mathbf{u}^{(i)}(\xi) \approx \mathbf{a}_0^{(i)} + \mathbf{a}_1^{(i)}\xi + \cdots + \mathbf{a}_{n_p-1}^{(i)}\xi^{n_p-1} \tag{24}$$

Similar to (15), the controls at the collocation points are

$$\begin{aligned} \mathbf{u}^{(i)}(\zeta) &= \underbrace{\begin{bmatrix} \mathbf{I}_{n_u} & \zeta_1\mathbf{I}_{n_u} & \ldots & \zeta_1^{n_p-1}\mathbf{I}_{n_u} \\ \mathbf{I}_{n_u} & \zeta_2\mathbf{I}_{n_u} & \ldots & \zeta_2^{n_p-1}\mathbf{I}_{n_u} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{I}_{n_u} & \zeta_{n_c}\mathbf{I}_{n_u} & \ldots & \zeta_{n_c}^{n_p-1}\mathbf{I}_{n_u} \end{bmatrix}}_{\boldsymbol{\zeta}_u} \underbrace{\begin{bmatrix} \mathbf{a}_0^{(i)} \\ \mathbf{a}_1^{(i)} \\ \vdots \\ \mathbf{a}_{n_p-1}^{(i)} \end{bmatrix}}_{\mathbf{a}_u^{(i)}} \\ &= \boldsymbol{\zeta}_u\boldsymbol{\theta}_u^{-1}\mathbf{b}_u^{(i)} = \boldsymbol{\phi}_u\mathbf{b}_u^{(i)} \end{aligned} \tag{25}$$

where the subscript $(\cdot)_u$ refers to expressions related to controls instead of states. Combining the controls at all collocation points into one vector $\hat{\mathbf{u}}(\zeta) = [\mathbf{u}^{(1)}(\zeta); \mathbf{u}^{(2)}(\zeta); \ldots; \mathbf{u}^{(i)}(\zeta); \ldots; \mathbf{u}^{(I)}(\zeta)]$, we can write

$$\hat{\mathbf{u}}(\zeta) = \mathbf{Z}_u\boldsymbol{\Theta}_u^{-1}\hat{\mathbf{b}}_u = \boldsymbol{\Phi}_u\hat{\mathbf{b}}_u \tag{26}$$

with similar notation as in Section III-A. Eq. (21) can now be rewritten to obtain

$$\boldsymbol{\Delta} = \boldsymbol{\Phi}'\hat{\mathbf{b}} - \frac{h}{2}[\hat{\mathbf{f}}_f + \hat{\mathbf{A}}(\boldsymbol{\Phi}\hat{\mathbf{b}} - \boldsymbol{\Phi}\hat{\mathbf{b}}^*)] + \hat{\mathbf{B}}\boldsymbol{\Phi}_u\hat{\mathbf{b}}_u = \mathbf{0} \tag{27}$$

Fig. 2 shows the difference between a nonlinear control interpolation (solid line) and a simple linear interpolation (dashed line) in the 7^{th}-order Legendre–Gauss–Lobatto method. The time instants t_i, t_{i+1}, t_{i+2}, and t_{i+3} are nodal points.

Remark 2: Due to numerical reasons, only LGL orders for which the interpolating polynomial of the control is of odd degree (i.e. when $n_p - 1$ is an odd number) can be considered. Results show that if LGL orders n are chosen such that $n_p - 1$ is even, the algorithm either fails to converge or finds solutions in which the thrust profile strongly suffers from oscillations.

IV. Homotopic Approach and Convex Programming

In this section, we present the sequential convex programming approach combined with a homotopic approach that considers continuation from the minimum-energy to the minimum-fuel problem to enhance the overall robustness. This method is similar to the one employed in literature to improve numerical convergence of indirect methods (see for example [27]).

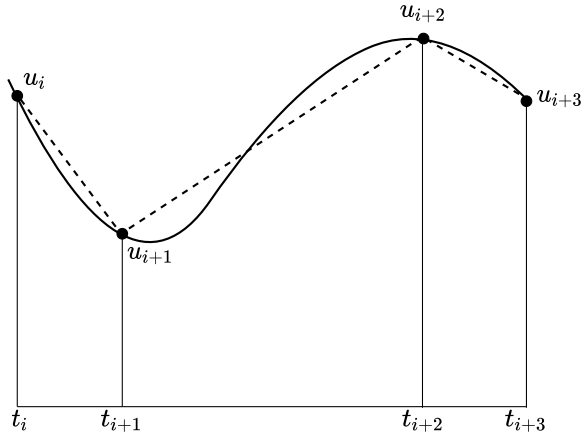


Fig. 2: Linear (dashed line) vs. nonlinear control interpolation (solid line) in the 7^{th}-order Legendre–Gauss–Lobatto method.

A. Sequential Convex Programming

The original nonconvex problem is solved by iteratively solving a sequence of convex subproblems. As the dynamical and thrust constraints have been linearized, the resulting problem may become infeasible even though the original problem is feasible. To avoid this *artificial infeasibility*, we add unconstrained slack variables $\boldsymbol{\nu}$ and η in (3a) and (3c) to obtain [7]

$$\dot{\mathbf{x}} = \mathbf{f}_f(\mathbf{x}^*) + \mathbf{A}(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) + \mathbf{B}\mathbf{u} + \boldsymbol{\nu} \tag{28}$$

$$0 \leq \tau \leq T_{\max} e^{-w^*}[1 - (w - w^*)] + \eta \tag{29}$$

Note that $\boldsymbol{\nu}$ and η must be zero at the end of the optimization process so that the constraints are satisfied. Therefore, the objective function J_0 in (2) is augmented with two terms [7]

$$J = J_0 + \sum_{i \in I_{eq}} \mu_i ||\boldsymbol{\nu}_i||_1 + \sum_{i \in I_{ineq}} \lambda_i \max(0, \eta_i) \tag{30}$$

where μ_i and λ_i are two sufficiently large parameters. I_{eq} and I_{ineq} are the set of equality and inequality constraints, respectively.

One key feature of the SCP algorithm is the trust-region mechanism to keep the solution close to the reference and thus the linearization valid. The ratio of the actual cost reduction $\Delta\phi_k = \phi_{k-1} - \phi_k$ to the predicted cost reduction $\Delta\hat{\phi}_k = \hat{\phi}_{k-1} - \hat{\phi}_k$ (with k denoting the current iteration) serves as a measure to decide whether the step is to be accepted or not. At each iteration, the merit functions ϕ_k and $\hat{\phi}_k$ are calculated as [37]

$$\phi_k = J_0 + \sum_{j \in I_{eq}} \mu_j ||\mathbf{h}_j||_1 + \sum_{j \in I_{ineq}} \lambda_j \max(0, g_j) \tag{31}$$

$$\hat{\phi}_k = J_0 + \sum_{j \in I_{eq}} \mu_j ||\boldsymbol{\nu}_j||_1 + \sum_{j \in I_{ineq}} \lambda_j \max(0, \eta_j) + \sum_{j \in I_{ineq}} \lambda_j \max(0, \sigma_j) \tag{32}$$

$\mathbf{h}_j$ and g_j are the constraint violations of the equality and inequality constraints, respectively, of the original, nonconvex problem. In contrast, (32) refers to the violations of the convex problem with $\sigma_j = \tau_{x,j}^2 + \tau_{y,j}^2 + \tau_{z,j}^2 - \tau_j^2$.

B. Homotopic Sequential Convex Programming

Instead of solving the minimum-fuel problem directly, a sequence of simpler and smoother problems is solved first in a homotopic approach. These solutions serve as initial guesses for the next subproblem where the complexity gradually increases until the original problem is eventually solved. One of the most common continuations is the energy-to-fuel homotopy which uses the objective function [24]

$$J_\gamma \equiv J(\gamma) = \int_{t_0}^{t_f} [(1 - \gamma)\tau + \gamma\tau^2] \, \mathrm{d}t \tag{33}$$

with some parameter $\gamma \in [0, 1]$ that defines the homotopic path from the minimum-energy ($\gamma = 1$) to the minimum-fuel ($\gamma = 0$) problem. τ is the acceleration magnitude defined in Section II. Note that the objective function in Eq. (33) is convex regardless of the value of γ, and thus it can be handled by convex solvers. We exploit the strategy developed in [38] to express the objective function in the standard form required by the second-order cone program solver we use.

In this work, we start solving the minimum-energy problem; then, γ is gradually decreased. Note that we change the homotopic parameter γ only when an SCP iteration is accepted. However, to speed up convergence, if the maximum nonlinear constraint violation $c_{\max}$ becomes less than $f_c\varepsilon_c$ (where ε_c is the convergence threshold and f_c is a large enough multiplication factor) before the maximum number of homotopic steps is reached, we set $\gamma = 0$.

The definition of the quantities ϕ_k and $\hat{\phi}_k$ in Eqs. (31) and (32) includes the objective function J_0. However, the values of J_0 cannot be compared while γ is still changing, simply because they refer to different problems. To avoid that the algorithm fails due to this reason, we substitute the term related to the objective function inside Eqs. (31) and (32) with a term associated with the final spacecraft mass. The aforementioned quantities are thus redefined as

$$\phi_k^F = \mu_{\mathrm{m}}(e^{w_0} - e^{w_f}) + \sum_{j \in I_{eq}} \mu_j ||\mathbf{h}_j||_1 + \sum_{j \in I_{ineq}} \lambda_j \max(0, g_j) \tag{34}$$

$$\hat{\phi}_k^F = \mu_{\mathrm{m}}(e^{w_0} - e^{w_f}) + \sum_{j \in I_{eq}} \mu_j ||\boldsymbol{\nu}_j||_1 + \sum_{j \in I_{ineq}} \lambda_j \max(0, \eta_j) + \sum_{j \in I_{ineq}} \lambda_j \max(0, \sigma_j) \tag{35}$$

where μ_{m} is a sufficiently large parameter such that $\mu_{\mathrm{m}} > 0$, and the superscript F is used to distinguish the expressions.

C. Adaptive Second-Order Trust-Region Radius Change

In standard trust-region methods such as in [4], [16], a step is rejected if $\rho_k \equiv \Delta\phi_k / \Delta\hat{\phi}_k < \rho_0$ because this indicates that

there is no (sufficiently large) progress. When a solution is accepted, the trust region is updated as follows:

$$R_{k+1} = \begin{cases} R_k/\alpha & \text{if} \quad \rho_0 \leq \rho^{(k)} < \rho_1 \\ R_k & \text{if} \quad \rho_1 \leq \rho^{(k)} < \rho_2 \\ \beta R_k & \text{if} \quad \rho^{(k)} \geq \rho_2 \end{cases} \tag{36}$$

where α and β are two constants greater than 1 and $0 < \rho_0 < \rho_1 < \rho_2 < 1$. Updating the trust-region radius like this often works well, but it is not particularly flexible for two reasons. First, the constants α and β must be selected by the user and are fixed during the optimization process. Secondly, the update only depends on the value of the parameter ρ at the current iteration k, without considering some potentially useful information from previous iterations.

We extend the work in [37] and propose an improved method where α and β can vary based on the values of ρ in the current k and previous iteration $k-1$. Introducing the new parameter $\delta = const. > 1$, we define the update mechanism of α and β as follows:

1. If $\rho^{(k)} \geq \rho_0$ and $\rho^{(k-1)} \geq \rho_0$ (that is, both the current and the previous steps are accepted), then $\beta_k = \delta\beta_{k-1}$ and $\alpha_k = \alpha_{k-1}/\delta$. Our rationale is that if two subsequent steps are accepted, the algorithm will benefit from a larger increase of the trust region in the next iteration.
2. If $\rho^{(k)} \geq \rho_0$ and $\rho^{(k-1)} < \rho_0$ (that is, the current step is accepted but the previous was not), then $\beta_k = \beta_{k-1}/\delta$ and $\alpha_k = \delta\alpha_{k-1}$. This suggests to stay closer to the current solution to avoid a rejected step in the next iteration.
3. If $\rho^{(k)} < \rho_0$ and $\rho^{(k-1)} \geq \rho_0$ (that is, the current step is rejected whereas the previous one was accepted), then $\alpha_k = \alpha_{k-1}$ and $\beta_k = \beta_{k-1}$. As the previous step was accepted, it is convenient to neither increase nor decrease the parameters. Note that this does not mean that the trust-region radius remains the same.
4. If $\rho^{(k)} < \rho_0$ and $\rho^{(k-1)} < \rho_0$ (that is, both the current and the previous steps are rejected), then $\alpha_k = \delta\alpha_{k-1}$ to stay closer to the reference solution and speed up convergence.

The parameters α and β increase the degrees of freedom of the algorithm and become part of the optimization process. Note that we impose $\alpha_{\min} \leq \alpha \leq \alpha_{\max}$ and $\beta_{\min} \leq \beta \leq \beta_{\max}$. Moreover, we select $\delta < \alpha_0, \beta_0$ (where the subscript $(\cdot)_0$ indicates the initial value) not to be too aggressive in the change of the parameters. The effectiveness of this approach is shown in Section V.

D. Algorithm

A flowchart is illustrated in Fig. 3, where the homotopic approach (bold blocks) and the adaptive trust region (dashed block) are integrated into the standard SCP method. The interested reader is referred to [7], [16], [37] for a more detailed description of the standard SCP algorithm. We focus instead on the integration of the homotopic approach.

At the beginning of each iteration, the value of γ defines the problem to be solved. For $0 < \gamma \leq 1$, the homotopic parameter is gradually decreased by $\Delta\gamma$. This process continues until the maximum constraint violation $c_{\max}$ is lower than some threshold $f_c\varepsilon_c$. When this threshold is reached, the algorithm sets $\gamma = 0$ to indicate that the minimum-fuel problem is to be solved in the next iteration.

V. Simulations and Results

The minimum-fuel transfer from Earth to asteroid Dionysus is considered in this work as it requires several revolutions around the Sun with considerable changes in the orbital elements. This problem is shown for the sake of comparison with previous works [7], [22]. We use the embedded conic solver (ECOS) [10] for all simulations, which are performed in MATLAB version R2020b on an Intel Core i7-10700 2.90 GHz desktop computer with 16 GB of RAM. We consider two-body dynamics without any additional perturbations, assume a constant specific impulse, and use the gravitational constant $\mu = 1.32712440018 \times 10^{20}$ m^3/s^2 and gravitational acceleration at sea level $g_0 = 9.80665$ m/s^2. The problem is scaled with the quantities given in Table I. Relevant parameters of the Earth to Dionysus transfer and the SCP algorithm are given in Table II and III, respectively. In particular, the values in Table III have been chosen in accordance with the typical ones reported in literature [7], [16]. Throughout this section, SCP refers to the standard algorithm with fixed α and β for the trust-region mechanism whereas SCP$_{\text{ATR}}$ uses the adaptive second-order trust-region update described in Section IV-C.

Despite some advanced methods to find a good initial guess exist in literature, such as the Finite Fourier Series (FFS) approach [39], [40] or the initial guess generator proposed in [41], we use a simple cubic interpolation to construct poor initial guesses and assess the robustness of our algorithm. This method requires the user to specify the number of revolutions N_{rev}. In this paper, the nominal case with $N_{\text{rev}} = 5$ is considered.

A. Comparison of Different LGL Orders

For different LGL orders, we compare SCP with SCP$_{\text{ATR}}$ in terms of the final mass of the spacecraft, number of iterations, computational time, convergence rate, and maximum error on the boundary conditions found when propagating the nonlinear dynamics with the obtained controls. This also serves as a validation of our algorithm when compared to results in the literature. We keep the number of nodes N approximately constant and close to 250. This value was chosen because it assures a good level of accuracy while maintaining an acceptable computational time. The results are presented in Table IV where n is the LGL order, *It.* is the number of iterations, *Cv.* is the convergence rate, and $\text{b}_{\max}$ is the aforementioned maximum error on the boundary conditions. Note that $\text{b}_{\max}$ is reported only once because the values are very similar for SCP and SCP$_{\text{ATR}}$. Fig. 6a compares the performance of SCP and SCP$_{\text{ATR}}$, where the results for different LGL orders have been averaged. Table V shows the correspondence between the selected LGL order n, the number of intervals I and nodes N. We consider only orders up to $n = 27$ because the accuracy of the solution decreases significantly for higher orders. We perform a robustness analysis with 100 cases and random perturbations on the nominal value of N_{rev} in the interval

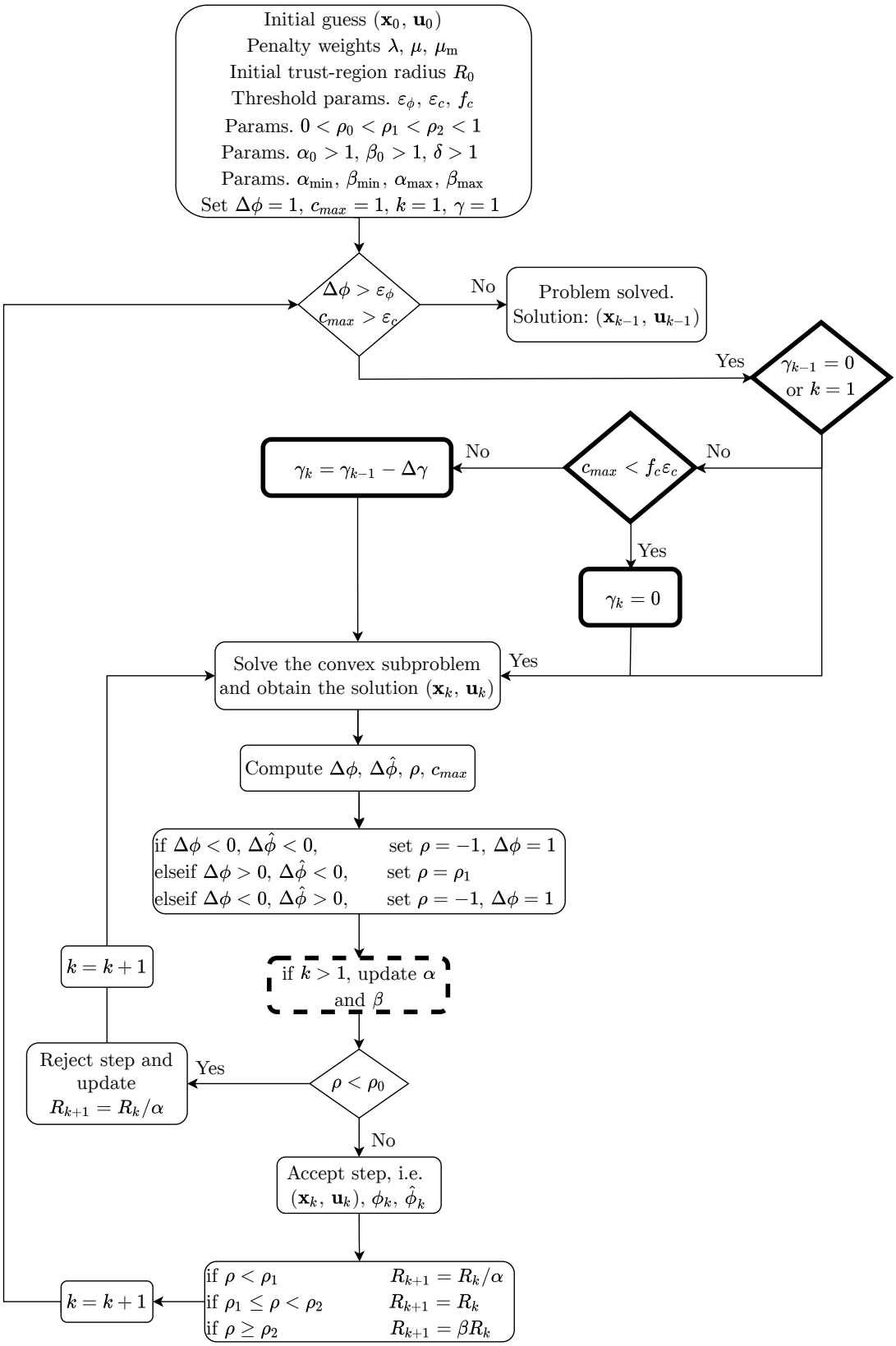


Fig. 3: Algorithm structure.

TABLE I: Physical quantities of the problem.

Physical Quantity	Unit	Normalization Factor	Value
Length	m	LU = 1 AU	$1.49597870 \times 10^{11}$
Velocity	m/s	VU $= \sqrt{\mu/\text{LU}}$	2.97846918×10^{4}
Time	s	TU $=$ LU/VU	5.02264286×10^{6}
Acceleration	m/s^2	ACU $=$ VU^2/LU	5.930083×10^{-3}
Mass	kg	MU = m_0	4000
Force	N	FU = $T_{\max}$	0.32

TABLE II: Parameters of Earth to Dionysus transfer.

Parameter	Unit	Value
$\mathbf{r}_0 = [r_x, r_y, r_z]_0^\top$	LU	$[-0.0243, 0.9833, -1.5117 \times 10^{-5}]^\top$
$\mathbf{v}_0 = [v_x, v_y, v_z]_0^\top$	VU	$[-1.0161, -0.0285, 1.6955 \times 10^{-6}]^\top$
m_0	MU	1
$\mathbf{r}_f = [r_x, r_y, r_z]_f^\top$	LU	$[-2.0406, 2.0518, 0.5543]^\top$
$\mathbf{v}_f = [v_x, v_y, v_z]_f^\top$	VU	$[-0.1423, -0.4511, 0.0189]^\top$
$T_{\max}$	FU	1
I_{sp}	TU	5.9730×10^{-4}
ToF	days	3534

TABLE III: Parameters of the algorithm.

Parameter	Value
Penalty weights λ, μ	500
Trust region r_0	100
Parameters ρ_0, ρ_1, ρ_2	$0.01, 0.25, 0.9$
Parameters α_0, β_0	1.4
Parameter δ	1.3
Upper bounds α_{max}, β_{max}	5.2
Lower bounds α_{min}, β_{min}	1.05
Feasibility threshold ε_c	10^{-5}
Optimality threshold ε_ϕ	10^{-2}
Parameter f_c	10^{3}
Parameter μ_m	30
Max. iterations	500

$[-10^{-1}, 10^{-1}]$. This way, the final boundary conditions are not satisfied, which deteriorates the convergence properties.

Remark 3: Each of the considered test cases corresponds to a perturbed cubic interpolation-based initial guess, which is used to assess the convergence properties of the proposed variants of our algorithm. The more the value of N_{rev} differs from the nominal one ($N_{\text{rev}} = 5$), the less accurate the initial guess. Fig. 5 shows the comparison of the nominal initial guess and a perturbed one.

In all simulations, the only stopping criterion is $c_{\max} \leq \varepsilon_c$, that is, the algorithm stops when the feasibility threshold is reached. This ensures a fair comparison because we are mainly interested in determining feasible solutions with respect to the nonlinear dynamics. Apparently, the enhanced trust-region method results in a considerable reduction of iterations and computational time regardless of the order of the polynomial; for the SCP$_{\text{ATR}}$ algorithm, both figures are only 30–40% of the correspondent ones of SCP. Yet, SCP has a higher convergence rate and is able to find slightly higher final masses $m(t_f)$ with respect to SCP$_{\text{ATR}}$. Note however that this behaviour can be improved by opting for a less aggressive update strategy, for example by selecting lower values for α_{max} and β_{max}, or by decreasing the value of the parameter δ. The LGL order does not seem to particularly affect convergence rates for the standard trust-region update, while higher values of n can significantly increase convergence for SCP$_{\text{ATR}}$ (up to $+10\%$). As expected, the CPU time increases for larger n because the matrices are denser. However, the accuracy of the solutions also increases with n, and therefore the number of nodes could be reduced for these cases to find a compromise between computational time and accuracy. Note that this is true only for orders up to $n = 23$, probably because high-order polynomials are not appropriate anymore to represent the states and controls. Compared to SCP where the number of iterations varies greatly over the different orders, SCP$_{\text{ATR}}$ required nearly the same amount of iterations (with the exception of the case $n = 27$).

Fig. 4a illustrates the three-dimensional trajectory for $n = 7$ in the nominal case with $N_{\text{rev}} = 5$; several revolutions are required to reach the target. The dashed lines represent the orbits of the Earth and Dionysus. The bang-bang control profile in Fig. 4b suggests that a locally optimal solution was found. In addition, the thrust profile, trajectory and final masses are very similar to the ones obtained with an indirect method (see Fig. 8 in [22] and also [23] where a final mass of 2718 kg is reported). Regardless of the LGL order and of the trust-region update mechanism, the obtained solutions

TABLE IV: Comparison of different LGL orders.

Alg. / Par.		SCP				SCP_{ATR}			
n	b_{max}	m_f (kg)	*It.*	Time (s)	*Cv.* (%)	m_f (kg)	*It.*	Time (s)	*Cv.* (%)
3	2.65e−2	2525	81.4	15.6	76.0	2330	33.3	6.6	58.0
7	3.68e−5	2534	88.8	52.0	74.0	2344	28.4	16.7	68.0
11	2.98e−4	2528	91.2	58.0	74.0	2424	26.5	16.8	59.0
15	5.83e−4	2522	95.9	78.8	74.0	2410	36.9	31.1	65.0
19	1.40e−3	2515	98.3	104.0	72.0	2371	27.4	29.0	62.0
23	4.60e−3	2537	97.9	134.8	72.0	2313	34.0	47.9	60.0
27	2.32e−2	2516	120.0	225.4	79.0	2284	54.9	104.1	66.0
Average	8.10e−3	2525	96.2	95.5	74.4	2354	34.5	36.0	62.6

TABLE V: Correspondence between LGL order n, intervals I and number of nodes N.

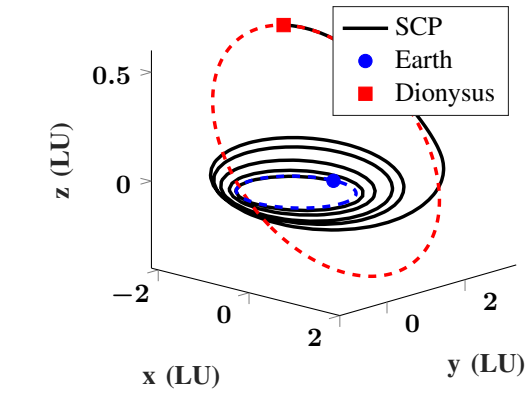
Parameter	**Value**						
n	3	7	11	15	19	23	27
I	249	83	50	36	28	23	19
N	250	250	251	253	253	254	248

look similar to the ones in Fig. 4. However, we noticed that when n is increased over 15–19 in case of SCP_{ATR}, the thrust profile contains few more points that are neither 0 nor T_{max} with respect to lower LGL orders. Nevertheless, constraints violation for high-orders is comparable to that of low ones.

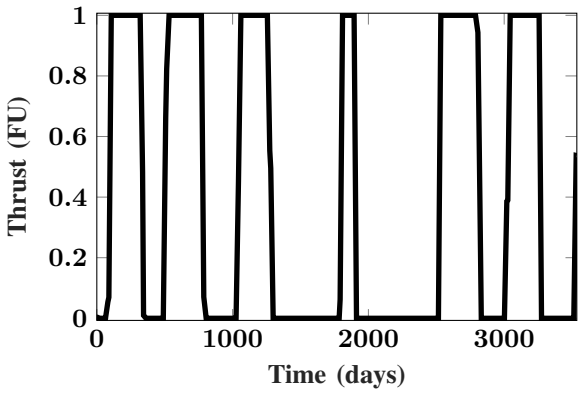
B. Homotopic Analysis

As the results in Section V-A are similar for different LGL orders, we choose $n = 3$ for all the subsequent analyses. Moreover, we select the SCP_{ATR} algorithm to understand whether our homotopic approach is able to increase its convergence up to the values of the standard trust-region update. We incorporate the homotopic approach of Section IV-B to increase the robustness against poor initial guesses. Our rationale is that solving the smooth minimum-energy problem is easier than solving the non-smooth minimum-fuel problem directly.

1) Comparison of Min-Fuel, Min-Energy, and Homotopic Approach: We consider the simple cubic interpolation approach with five revolutions as the nominal trajectory for the initial guess. We perform a robustness analysis with 100 cases and the same perturbations as in Section V-A. A maximum number of $s = 10$ equally spaced homotopic steps is used to gradually decrease γ from 1 (minimum-energy) to 0 (minimum-fuel). Recall from Section IV-B, however, that the algorithm switches to the minimum-fuel problem as soon as the condition $c_{max} < f_c \varepsilon_c$ is met and therefore, the actual number may be smaller. The results are presented in Fig. 6b and Table VI where the average final mass, iterations, CPU time, and convergence are compared. The error bars in Fig. 6b indicate minimum and maximum values, respectively. Note that the results for the minimum-fuel problem are the same as in Table IV.



(a) Minimum-fuel transfer trajectory.



(b) Thrust profile.

Fig. 4: Transfer trajectory and thrust profile for n = 7.

Remarkably, solving the smoother problem results in a considerably higher success rate: 71 cases converged for the minimum-energy, and 68 for the homotopic minimum-fuel problem, while the fuel-optimal one converged in only 58 cases. The final masses obtained with the homotopic fuel-optimal approach are slightly higher with respect to the ones obtained with the minimum-fuel problem. As expected, the minimum-energy case found lower final masses instead. Moreover, the number of iterations is also larger for the minimum-fuel case. Note that the average number of iterations

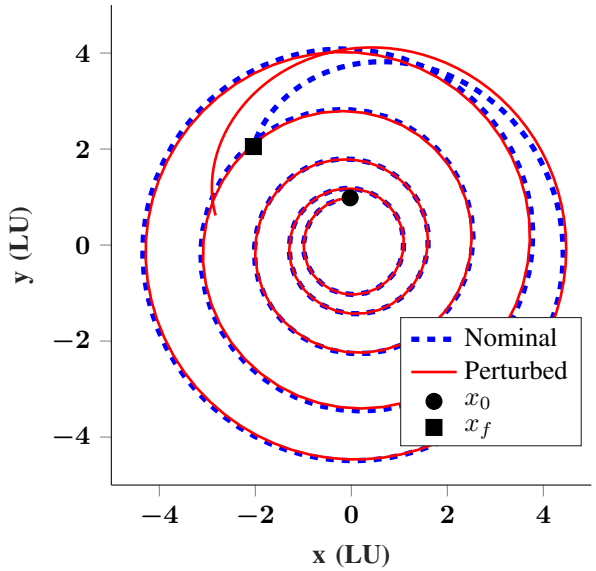
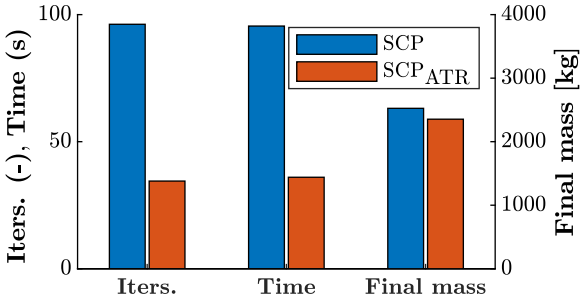
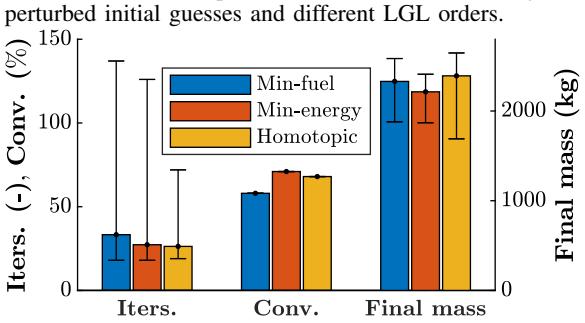


Fig. 5: Nominal initial guess ($N_{\text{rev}} = 5$) and perturbed initial guess ($N_{\text{rev}} = 5.0913$).



(a) Performance comparison of SCP and SCP$_{\text{ATR}}$ algorithms with perturbed initial guesses and different LGL orders.

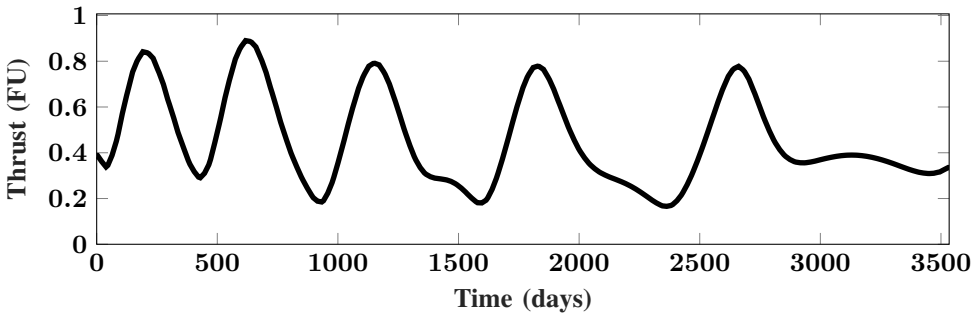


(b) Performance comparison of SCP$_{\text{ATR}}$ for minimum-fuel, minimum-energy and homotopic problems with perturbed initial guesses.

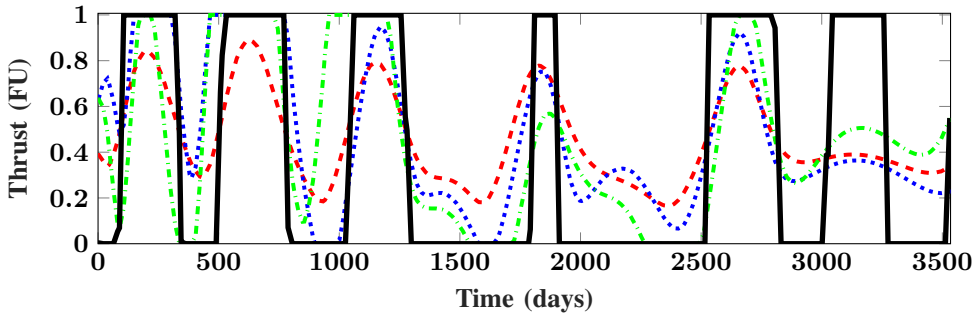
Fig. 6: Statistics for different problems and versions of the algorithm.

for the minimum-energy and the homotopic minimum-fuel problem are very similar instead. Fig. 7 illustrates the thrust profile obtained with the minimum-energy problem and a comparison of thrust profiles associated with different values of the smoothing parameter γ, respectively. It can be observed that the minimum-energy thrust profile is perfectly smooth, and that the smoothness decreases as the homotopic parameter approaches zero. Note that the figures show the profiles obtained by solving the associated problems without any homotopy.

2) Influence of Homotopic Steps and Distribution: We investigate the influence of the number of homotopic steps and their distribution on the performance of our approach. In addition to equidistant steps, two logarithmic distributions with higher density close to $\gamma = 1$ (Fig. 8a) and close to $\gamma = 0$ (Fig. 8b) are considered. For each of them, four analyses with different numbers of homotopic steps are carried out. In particular, we use $s = 10$, $s = 50$, $s = 100$ and $s = 500$. The simulations consider again the same 100 perturbed test cases already used to perform the analyses described in the previous sections. Tables VII, VIII, IX show the results obtained with the equidistant and the two logarithmic distributions of the steps, respectively. In general, neither the distribution of the steps nor their number particularly influence the performance of the algorithm. In fact, the values of the average final masses are practically identical for the three different distributions, as well as average iterations and computational time. As expected, the logarithmic case with denser distribution of steps close to $\gamma = 1$ performed best in terms of converged cases, followed by the linear case. Unexpectedly, fewer steps imply more converged cases for the logarithmic distribution with higher density of steps close to $\gamma = 0$. Finally, it is remarkable that in the considered simulations, all the cases show better properties in terms of converged cases, iterations and CPU time with respect to the simple minimum-fuel problem (see Table VI). This confirms the effectiveness of the homotopic approach.

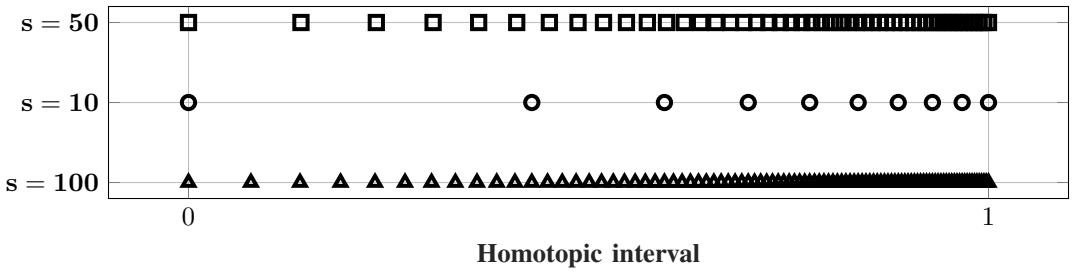


(a) Minimum-energy thrust profile.

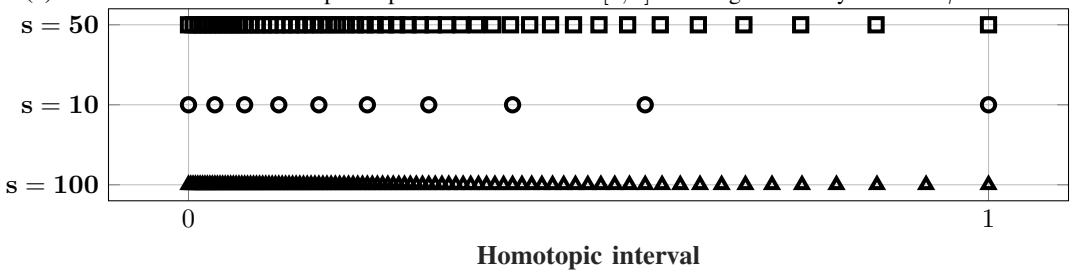


(b) Thrust profiles obtained with $\gamma = 1$ (dashed line), $\gamma = 0.7$ (dotted line), $\gamma = 0.3$ (dash-dotted line), and $\gamma = 0$ (solid line).

Fig. 7: Comparison of thrust profiles for different problems with $N_{\text{rev}} = 5$.



(a) Distribution of the homotopic steps inside the interval $[0, 1]$ with higher density close to $\gamma = 1$.



(b) Distribution of the homotopic steps inside the interval $[0, 1]$ with higher density close to $\gamma = 0$.

Fig. 8: Logarithmic distribution of the homotopic steps inside the interval $[0, 1]$.

TABLE VI: Comparison of SCP_{ATR} for minimum-fuel, minimum-energy and homotopic minimum-fuel problems with perturbed initial guesses.

Problem / Element	Min-fuel	Min-energy	Homotopic
$m(t_f)$ (kg)	2330	2214	2392
Iterations	33.3	27.3	26.3
CPU Time (s)	6.6	6.4	5.9
Convergence (%)	58	71	68

TABLE VII: Convergence analysis of the Homotopic SCP_{ATR} algorithm varying s with linear distribution of homotopic steps.

Element / Nr. steps s	$m(t_f)$ (kg)	*It.*	**Time** (s)	*Cv.* (%)
10	2392	26.3	5.9	68.0
50	2385	24.2	5.3	68.0
100	2385	24.9	5.4	67.0
500	2385	25.7	5.5	70.0
Average	2387	25.3	5.3	68.3

TABLE VIII: Convergence analysis of the Homotopic SCP_{ATR} algorithm varying s with logarithmic distribution of homotopic steps and higher density near $\gamma = 1$.

Element / Nr. steps s	$m(t_f)$ (kg)	*It.*	**Time** (s)	*Cv.* (%)
10	2391	23.3	5.1	68
50	2393	25.1	5.4	70
100	2396	25.8	5.5	70
500	2368	26.1	5.5	69
Average	2387	25.1	5.4	69.3

TABLE IX: Convergence analysis of the Homotopic SCP_{ATR} algorithm varying s with logarithmic distribution of homotopic steps and higher density near $\gamma = 0$.

Element / Nr. steps s	$m(t_f)$ (kg)	*It.*	**Time** (s)	*Cv.* (%)
10	2355	25.3	5.8	70.0
50	2398	28.0	6.2	68.0
100	2406	25.5	5.7	66.0
500	2404	24.1	5.2	64.0
Average	2391	25.7	5.7	67.0

VI. Conclusions

In this paper, a robust algorithm based on convex optimization is proposed to solve the low-thrust minimum-fuel space trajectory optimization problem. An arbitrary-order Gauss–Lobatto discretization method with nonlinear control interpolation is exploited to improve the performance of the standard SCP. Results show that higher-order LGL discretization can increase the convergence rate of the simple Hermite–Simpson rule and the accuracy of the results, at the expense of a higher computational time. Moreover, the newly-developed adaptive trust-region radius update shows superior performance in terms of computational time with respect to the standard approach. Finally, the combination of a homotopic approach with convex programming improves convergence of the minimum-fuel space trajectory optimization problem. Our assumption that the smooth minimum-energy problem is easier to solve than the minimum-fuel problem directly was confirmed in numerical simulations. Overall, the proposed algorithm represents a promising alternative to standard nonlinear programming methods in scenarios like deep-space cruise where robustness

and convergence are more important than optimality. The rapid speed makes our method an ideal choice for preliminary studies and also real-time applications.

ACKNOWLEDGMENT

This research is part of EXTREMA, a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 864697).

REFERENCES

[1] K. Woellert, P. Ehrenfreund, A. J. Ricco, and H. Hertzfeld, "Cubesats: Cost-effective science and technology platforms for emerging and developing nations," *Adv. Space Res.*, vol. 47, no. 4, pp. 663–684, Feb. 2011.

[2] R. Walker, D. Binns, C. Bramanti, M. Casasco, P. Concari, D. Izzo, D. Feili, P. Fernandez, J. G. Fernandez, P. Hager, D. Koschny, V. Pesquita, N. Wallace, I. Carnelli, M. Khan, M. Scoubeau, and D. Taubert, "Deep-space cubesats: Thinking inside the box," *Astron. Geophys.*, vol. 59, no. 5, pp. 5–24, Oct. 2018.

[3] R. Chai, A. Tsourdos, A. Savvaris, S. Chai, Y. Xia, and C. P. Chen, "Review of advanced guidance and control algorithms for space/aerospace vehicles," *Prog. Aerosp. Sci.*, vol. 122, p. 100696, Mar. 2021.

[4] Z. Wang and M. J. Grant, "Minimum-fuel low-thrust transfers for spacecraft: A convex approach," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 5, pp. 2274–2290, Mar. 2018.

[5] J. T. Betts, "Survey of numerical methods for trajectory optimization," *J. Guid. Control Dyn.*, vol. 21, no. 2, pp. 193–207, Mar. 1998.

[6] B. A. Conway, "A survey of methods available for the numerical optimization of continuous dynamic systems," *J. Optim. Theory Appl.*, vol. 152, no. 2, p. 271–306, Sep. 2012.

[7] C. Hofmann and F. Topputo, "Rapid low-thrust trajectory optimization in deep space based on convex programming," *J. Guid. Control Dyn.*, vol. 44, no. 7, pp. 1379–1388, Apr. 2021.

[8] B. Pan, P. Lu, X. Pan, and Y. Ma, "Double-homotopy method for solving optimal control problems," *J. Guid. Control Dyn.*, vol. 39, no. 8, pp. 1706–1720, Jun. 2016.

[9] X. Liu, P. Lu, and B. Pan, "Survey of convex optimization for aerospace applications," *Astrodyn.*, vol. 1, no. 1, pp. 23–40, Sep. 2017.

[10] A. Domahidi, E. Chu, and S. Boyd, "Ecos: An socp solver for embedded systems," in *Proc. Eur. Control Conf.*, Zurich, Switzerland, 2013, pp. 3071–3076.

[11] S. Michael, C. A. Pascucci, D. Dueri, and B. Açıkmeşe, "Convexification and real-time on-board optimization for agile quad-rotor maneuvering and obstacle avoidance," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Vancouver, BC, Canada, Sep. 2017, pp. 4862–4868.

[12] M. W. Harris and B. Açıkmeşe, "Lossless convexification for a class of optimal control problems with linear state constraints," in *Proc. IEEE Conf. Decision Control*, Florence, Italy, Dec. 2013, pp. 7113–7118.

[13] B. Açıkmeşe, J. M. Carson, and L. Blackmore, "Lossless convexification of nonconvex control bound and pointing constraints of the soft landing optimal control problem," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 6, pp. 2104–2113, Nov. 2013.

[14] X. Liu, Z. Shen, and P. Lu, "Exact convex relaxation for optimal flight of aerodynamically controlled missiles," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 4, pp. 1881–1892, May 2016.

[15] Y. Mao, M. Szmuk, and B. Açıkmeşe, "Successive convexification of non-convex optimal control problems and its convergence properties," in *Proc. IEEE Conf. Decision Control*, Las Vegas, NV, USA, Dec. 2016, pp. 3636–3641.

[16] Y. Mao, M. Szmuk, X. Xu, and B. Açıkmeşe, "Successive convexification: A superlinearly convergent algorithm for non-convex optimal control problems," Preprint, submitted Feb. 2019, https://arxiv.org/abs/1804.06539.

[17] R. Bonalli, A. Cauligi, A. Bylard, and M. Pavone, "Gusto: Guaranteed sequential trajectory optimization via sequential convex programming," in *Proc. IEEE Conf. on Robot. and Autom.*, Montreal, QC, Canada, May 2019, pp. 6741–6747.

[18] M. Sagliano, "Generalized *hp* pseudospectral-convex programming for powered descent and landing," *J. Guid. Control Dyn.*, vol. 42, no. 7, pp. 1562–1570, Apr. 2019.

[19] X. Liu, Z. Shen, and P. Lu, "Entry trajectory optimization by second-order cone programming," *J. Guid. Control Dyn.*, vol. 39, no. 2, pp. 227–241, Aug. 2016.

[20] R. Chai, A. Tsourdos, A. Savvaris, S. Chai, and Y. Xia, "Trajectory planning for hypersonic reentry vehicle satisfying deterministic and probabilistic constraints," *Acta Astronaut.*, vol. 177, pp. 30–38, Dec. 2020.

[21] Y. Song and S. Gong, "Solar-sail deep space trajectory optimization using successive convex programming," *Astrophys. Space Sci.*, vol. 364, no. 106, Jul. 2019.

[22] E. Taheri, I. Kolmanovsky, and E. Atkins, "Enhanced smoothing technique for indirect optimization of minimum-fuel low-thrust trajectories," *J. Guid. Control Dyn.*, vol. 39, no. 11, pp. 2500–2511, Sep. 2016.

[23] E. Taheri and J. L. Junkins, "Generic smoothing for optimal bang-off-bang spacecraft maneuvers," *J. Guid. Control Dyn.*, vol. 41, no. 11, pp. 2467–2472, Sep. 2018.

[24] F. Jiang, H. Baoyin, and J. Li, "Practical techniques for low-thrust trajectory optimization with homotopic approach," *J. Guid. Control Dyn.*, vol. 35, no. 1, pp. 245–258, Aug. 2012.

[25] C. Zhang, F. Topputo, F. Bernelli-Zazzera, and Y.-S. Zhao, "Low-thrust minimum-fuel optimization in the circular restricted three-body problem," *J. Guid. Control Dyn.*, vol. 38, no. 8, pp. 1501–1510, Mar. 2015.

[26] J. F. Li, Z. Che, and H. N. Li, "Applications of homotopic approach in variable-specific-impulse low-thrust trajectory optimization," in *Proceedings of IEEE Chinese Guidance, Navigation and Control Conference*, Aug. 2014, pp. 2100–2104.

[27] E. Taheri, E. M. Atkins, and I. Kolmanovsky, "Performance comparison of smoothing functions for indirect optimization of minimum-fuel low-thrust trajectories," in *Proceedings of Space Flight Mechanics Meeting, AIAA SciTech Forum, AIAA Paper 2018-0214*, Kissimmee, FL, USA, Jan. 2018.

[28] R. Bertrand and R. Epenoy, "New smoothing techniques for solving bang-bang optimal control problems numerical results and statistical interpretation," *Optim. Control Appl.*, vol. 23, no. 4, pp. 171–197, Aug. 2002.

[29] X. Pan and B. Pan, "Practical homotopy methods for finding the best minimum-fuel transfer in the circular restricted three-body problems," *IEEE Access*, vol. 8, pp. 47 845–47 862, Feb. 2020.

[30] F. Topputo and C. Zhang, "Survey of direct transcription for low-thrust space trajectory optimization with applications," *Abstr. Appl. Anal.*, vol. 2014, Jun. 2014.

[31] M. Sagliano and E. Mooij, "Optimal drag-energy entry guidance via pseudospectral convex optimization," *Aerosp. Sci. Technol.*, vol. 117, p. 106946, Jul. 2021.

[32] M. Szmuk, T. Reynolds, and B. Açıkmeşe, "Successive convexification for real-time six-degree-of-freedom powered descent guidance with state-triggered constraints," *J. Guid. Control Dyn.*, vol. 43, no. 8, pp. 1399–1413, Jun. 2020.

[33] P. Williams, "Hermite–legendre–gauss–lobatto direct transcription in trajectory optimization," *J. Guid. Control Dyn.*, vol. 32, no. 4, pp. 1392–1395, Aug. 2009.

[34] D. Malyuta, T. Reynolds, M. Szmuk, M. Mesbahi, B. Açıkmeşe, and J. M. Carson, "Discretization performance and accuracy analysis for the rocket powered descent guidance problem," in *Proceedings of AIAA SciTech Forum, AIAA Paper 2019-0925*, San Diego, CA, USA, Jan. 2019.

[35] A. Ronveaux and J. Mawhin, "Rediscovering the contributions of rodrigues on the representation of special functions," *Expo. Math.*, vol. 23, no. 4, pp. 361–369, Dec. 2005.

[36] C. Zhang, F. Topputo, F. Bernelli-Zazzera, and Y.-S. Zhao, "An exploration of numerical methods for low-thrust trajectory optimization in n-body models," in *Proceedings of International Astronautical Congress*, Beijing, China, Sep. 2013, pp. 23–27.

[37] C. Hofmann and F. Topputo, "Toward on-board guidance of low-thrust spacecraft in deep space using sequential convex programming," in *Proceedings of AAS/AIAA Space Flight Mechanics Meeting AAS Paper 21-350*, Feb. 2021, pp. 1–19.

[38] ——, "Pseudospectral convex low-thrust trajectory optimization in a high-fidelity model," in *Proceedings of AAS/AIAA Astrodynamics Specialist Conference AAS Paper 21-678*, Aug. 2021, pp. 1–19.

[39] W. Zhou, H. Li, H. Wang, and R. Ding, "Low-thrust trajectory design using finite fourier series approximation of pseudoequinoctial elements," *Int. J. Aerosp. Eng.*, vol. 2019, Nov. 2019.

[40] E. Taheri and O. Abdelkhalik, "Shape-based approximation of constrained low-thrust space trajectories using fourier series," *J. Spacecr. Rockets*, vol. 49, no. 3, pp. 535–545, Jun. 2012.

[41] R. Chai, A. Savvaris, A. Tsourdos, S. Chai, and Y. Xia, "Trajectory optimization of space maneuver vehicle using a hybrid optimal control solver," *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 467–480, Feb. 2019.

Andrea Carlo Morelli is a PhD student at the Deep-space Astrodynamics Research and Technology (DART) group, Politecnico di Milano, working on autonomous guidance of interplanetary CubeSats in the context of the ERC-funded project EXTREMA. He obtained his Bachelor of Science in Aerospace Engineering from Politecnico di Milano in 2018. He has been a student of the merit program Alta Scuola Politecnica and earned his Master of Science in Space Engineering at Politecnico di Milano in 2020. He spent seven months at ISAE-Supaero in Toulouse in the context of a European exchange program.

Christian Hofmann is a PhD student in Aerospace Engineering and currently works on the autonomous guidance of interplanetary CubeSats. He holds a Master of Science in Aerospace Engineering from the Technical University of Braunschweig and a Bachelor of Science in Aerospace Computer Science from the University of Würzburg in Germany. He worked as a research assistant at the German Aerospace Center (DLR) before he joined the DART group in November 2018. His research interests include astrodynamics, nonlinear optimal control, interplanetary CubeSats, space trajectory optimization and computational guidance and control.

Francesco Topputo is a Full Professor of Space Systems at Politecnico di Milano, Italy, and holds a position as Visiting Professor at TU Delft, The Netherlands. His core research activities involve spacecraft flight dynamics, interplanetary CubeSat mission and system design, autonomous guidance, navigation, and control. Dr. Topputo is an ERC laureate (CoG 2019) and has been PI in 14 research projects. He leads the DART group at Politecnico di Milano. He has published 55 peer-reviewed articles in international journals and over 180 works in total.