CEAS-GNC-2024-071

EuroGNC Navigation Guidance Control CEAS Bristol, UK June 11th-13th 2024
University of BRISTOL ROYAL AERONAUTICAL SOCIETY AIAA

Analysis of the Covector Mapping Principle using Non-Complete Methods

Silvia Busi Research Associate, University of Stuttgart, Institute of Flight Mechanics and Controls, 70569, Stuttgart, Germany. silvia.busi@ifr.uni-stuttgart.de

Torbjørn Cunis Lecturer, University of Stuttgart, Institute of Flight Mechanics and Controls, 70569, Stuttgart, Germany. torbjoern.cunis@ifr.uni-stuttgart.de

Francesco Topputo Professor, Politecnico di Milano, Department of Aerospace Science and Technology, 20156, Milan, Italy. francesco.topputo@polimi.it

Walter Fichter Professor, University of Stuttgart, Institute of Flight Mechanics and Controls, 70569, Stuttgart, Germany. walter.fichter@ifr.uni-stuttgart.de

ABSTRACT

This paper focuses on the application of the Covector Mapping Principle (CMP) to low-thrust trajectory optimization problems to analyze the performances of the algorithm when non-complete methods are employed. Non-complete methods are those numerical schemes that do not provide an analytical mapping between the dual variables of direct and indirect approaches. However, since these approaches stem from the same original continuous-time Optimal Control Problem, it is intuitive that a link between them should exist. It is investigated and applied to an Energy Optimal Problem to overcome the disadvantages characterizing direct and indirect methods when used separately in order to improve mission planning. The same problem is solved with a global Pseudospectral method, belonging to the class of complete schemes, to compare the results and verify if and to what extent the CMP can be exploited with non-complete methods. The results show the mapping to be successful even without an analytical law, allowing to obtain both accuracy and robustness, and thus overcoming the disadvantages of the two main classes of numerical methods. The performances obtained with the complete method are better due to both its fast convergence properties and the availability of an analytical mapping.

Keywords: Low-Thrust Trajectory Optimization; Nonlinear Optimal Control; Covector Mapping Principle; Direct Methods

Nomenclature

CMP	=	Covector Mapping Principle
CMT	=	Covector Mapping Theorem
EO	=	Energy Optimal
LGR	=	Legendre-Gauss-Radau
NLP	=	Nonlinear Programming Problem
OCP	=	Optimal Control Problem

ODE = Ordinary Differential Equation
PS = Pseudospectral
RK = Runge Kutta
ToF = Time of Flight
TPBVP = Two Point Boundary Value Problem

1 Introduction

The Optimal Control Problem (OCP) consists in determining the input, namely the controls, to a dynamical system in such a manner that a specified performance index is optimized while a set of constraints is satisfied. It is used to address the problem of mission planning, which is performed offline and plays a crucial role in reaching the goal of a space mission while keeping the cost reasonable. Therefore, robustness of the algorithm and accuracy of the solution are important for the mission objectives. In recent years, electric propulsion is increasingly gaining attention in this field, as it is more efficient than classical chemical propulsion, guaranteeing a specific impulse approximately ten times higher [1]. Propellant mass is consequently saved, allowing for heavier payloads to be embarked. In addition, both launch windows are extended and control abilities are more flexible and precise. However, this technology provides a low level of thrust (in the order of mN), leading to a rather difficult trajectory optimization process, since it requires the determination of a continuous control law. An analytical solution for this problem does not exist because of the high complexity it embeds. Therefore, numerical techniques have been developed and are continuously improved. They are commonly classified as either direct or indirect methods [2]. The first approach discretizes and then optimizes the problem, and is characterized by a larger convergence basin, but does not guarantee solution optimality. The latter, instead, firstly derives the optimality conditions, and then solves the problem numerically. It is considered as more accurate, but difficult to initialize. Despite these intrinsic features, the methods stem from the same original continuous-time OCP, and therefore it seems intuitive that a connection between them exists [3]. The Covector Mapping Principle (CMP) works on the dual variables of the methods to exploit this link with the aim of overcoming the limitations and providing an algorithm that is both accurate and robust. Researchers have demonstrated that only few methods, denoted as *complete* [4], provide an analytical link through a Covector Mapping Theorem (CMT), and so this paper analyzes both if and to what extent the CMP can be exploited with *non-complete* methods. A different approach is presented in [5], where dual variables of the indirect formulation are approximated starting from the results obtained using a direct approach.

In particular, the aim of this work is to improve mission planning by combining the two main classes of methods through the CMP with non-complete methods. It is proved both whether costates pass on the information of optimality to direct methods, being an initial guess for Lagrange multipliers, and also if robustness and flexibility of direct techniques can be exploited to solve the problem and to provide a good initial guess for the indirect costates. Hermite-Simpson scheme [2] is used as direct non-complete collocation method, while a comparison is done by employing a complete Pseudospectral (PS) method with Legendre-Gauss-Radau (LGR) collocation [6, 7]. A simple shooting algorithm with Runge-Kutta (RK) 7-8 integration scheme is adopted within the indirect formulation [8]. Performances are analyzed and compared by solving a two-body low-thrust Energy Optimal (EO) problem, showing the possibility of exploiting the principle with non-complete methods.

The remainder of the paper is organized as follows: section 2 introduces the trajectory optimization problem and the numerical methods traditionally used for the solution. The CMP is presented in section 3, where also the mapping between dual variables for non-complete methods is provided. Section 4 analyzes the results of the simulation, and eventually section 5 presents the conclusions along with some open points.

2 The Trajectory Optimization Problem

Without loss of generality, consider the EO trajectory optimization problem in Lagrange form. It consists in determining the control history $\boldsymbol{u}(t)$ that minimizes a given functional J, denoted as *performance index*, while satisfying dynamic constraints $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t)$, and boundary conditions $\boldsymbol{\phi}(\boldsymbol{x}(t_0), t_0, \boldsymbol{x}(t_f), t_f) = \boldsymbol{0}$. This results in the following mathematical statement:

$$\text{minimize} \qquad J = \int_{t_0}^{t_f} \mathcal{L}(\boldsymbol{x}(t), \boldsymbol{u}(t), t)dt, \tag{1a}$$

$$\text{subject to} \qquad \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t), \qquad \forall t \in [t_0, t_f], \tag{1b}$$

$$\boldsymbol{\phi}(\boldsymbol{x}(t_0), t_0, \boldsymbol{x}(t_f), t_f) = \boldsymbol{0} \tag{1c}$$

where $\mathcal{L}(\boldsymbol{x}(t), \boldsymbol{u}(t), t)$ represents the running cost [9]. In particular, for the two-body low-thrust optimization problem expressed in Cartesian coordinates, the states correspond to position $\boldsymbol{r}$, velocity $\boldsymbol{v}$, and spacecraft mass m. The cost function for the EO problem and the continuous equations of motion can be written as

$$J = \int_{t_0}^{t_f} \frac{T_{\max}}{I_{\text{sp}} g_0} u^2 dt, \tag{2}$$

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{r}} \\ \dot{\boldsymbol{v}} \\ \dot{m} \end{bmatrix} = \begin{bmatrix} \boldsymbol{v} \\ -\dfrac{\mu_\odot}{r^3}\boldsymbol{r} + \dfrac{uT_{\max}}{m}\boldsymbol{\alpha} \\ -\dfrac{uT_{\max}}{I_{\text{sp}} g_0} \end{bmatrix}. \tag{3}$$

where $\mu_\odot$ is the Sun gravitational constant and g_0 the acceleration of gravity at sea level on Earth. In this work, both maximum thrust $T_{\max}$ and specific impulse I_{sp} are considered as constant. The control variables are the thrust magnitude $u \in [0, 1]$, and the thrust direction unit $\boldsymbol{\alpha}$, that embeds in-plane α and out-of-plane β angles,

$$\boldsymbol{\alpha} = \begin{bmatrix} \sin(\alpha)\cos(\beta) & \cos(\alpha)\cos(\beta) & \sin(\beta) \end{bmatrix}^T. \tag{4}$$

As already mentioned in section 1, an analytical solution to low-thrust trajectory optimization problems does not exist without a sufficient level of assumptions, and therefore they are solved numerically using the methods introduced in the next section.

2.1 Numerical methods

The numerical approaches adopted to solve trajectory optimization problems are divided into two main categories, namely direct and indirect methods [2].

Indirect methods

In an indirect method, calculus of variations leads to the derivation of first-order necessary conditions for optimality, also denoted as Euler-Lagrange equations [10]. The augmented cost functional $\hat{J}$ is introduced through the dual variables $\boldsymbol{\nu}$ linked to the boundary conditions and through the vector of costate multipliers of the dynamics, $\boldsymbol{\lambda}(t)$, resulting in

$$\hat{J} = \boldsymbol{\nu}^T \boldsymbol{\phi}(\boldsymbol{x}(t_0), t_0, \boldsymbol{x}(t_f), t_f) + \int_{t_0}^{t_f} [\mathcal{L}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) + \boldsymbol{\lambda}^T(t)(\boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) - \dot{\boldsymbol{x}}(t))]dt. \tag{5}$$

The problem consists in formulating the necessary conditions for a stationary point of $\hat{J}$. Therefore, its first derivative $\delta\hat{J}$ is set to zero. In order to obtain the necessary conditions in a compact form, the Hamiltonian H of the problem is defined as

$$H(\boldsymbol{x}(t), \boldsymbol{\lambda}(t), \boldsymbol{u}(t), t) = \mathcal{L}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) + \boldsymbol{\lambda}^T(t)\boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t), \tag{6}$$

and the resulting first-order necessary optimality conditions read

$$\dot{\boldsymbol{x}} = H_\lambda, \qquad \dot{\boldsymbol{\lambda}} = -H_x, \qquad \mathbf{0} = H_u, \tag{7}$$

where the subscripts denote partial derivation. The first and second equations correspond respectively to dynamic constraints and costate dynamics, also denoted as adjoint equation, while the third one is an algebraic equation for the controls. A more general expression for the latter corresponds to *Pontryagin's Minimum Principle* [11]

$$\boldsymbol{u}(t) = \underset{\boldsymbol{u}\in U}{\operatorname{argmin}} \quad H(\boldsymbol{x}(t), \boldsymbol{\lambda}(t), \boldsymbol{u}(t), t), \tag{8}$$

where U is the set of feasible controls [9]. The differential-algebraic system in Equation (7) shall be solved together with the final boundary conditions and the following transversality condition

$$\boldsymbol{\lambda}(t_f) = [\boldsymbol{\nu}^T\boldsymbol{\phi}_x]_{t=t_f}. \tag{9}$$

Euler-Lagrange equations represent necessary conditions [12], and, together, they form a two-point boundary value problem (TPBVP), which is solved using a numerical method.

Direct methods

Direct methods, instead, transcribe the original OCP into a mathematical programming problem without deriving the first-order necessary optimality conditions. This process involves the approximation of control and/or state variables over a discretized time grid [2]. The resulting problem is then optimized using Nonlinear Programming (NLP) solvers, which are based on Newton scheme, and therefore are denoted as gradient-based methods.
This work employs direct collocation method, whose features are now presented. The time interval $[t_0, t_f]$ is discretized into a finite sequence of N successive time instants t_i, and both states and controls are approximated on this time grid. This step can be summarized as

$$[t_0, t_f] \quad \Rightarrow \quad t_i, \quad i = 1, ..., N, \qquad \text{where} \quad t_1 = t_0, \quad t_N = t_f, \tag{10}$$

$$\boldsymbol{x}_i = \boldsymbol{x}(t_i), \quad \boldsymbol{u}_i = \boldsymbol{u}(t_i). \tag{11}$$

Then, the dynamics constraints in Equation (1b) are replaced by defect constraints and imposed at t_i. Their formulation depends on the adopted numerical scheme. The most extensively used are Hermite-Simpson scheme [2], high-order Gauss-Lobatto quadrature schemes [13], and PS methods [6]. The defect constraints and the boundary constraints in Equation (1c) are grouped together within a vector $\boldsymbol{c}(\boldsymbol{y})$, where $\boldsymbol{y}$ is the vector of decision variables

$$\boldsymbol{y} = [\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_N^T, \boldsymbol{u}_1^T, \ldots, \boldsymbol{u}_N^T]^T = [\boldsymbol{X}^T, \boldsymbol{U}^T]^T. \tag{12}$$

The optimization process aims at finding the pairs $\boldsymbol{x}_i, \boldsymbol{u}_i$ that minimize the objective function F while satisfying the constraints $\boldsymbol{c}(\boldsymbol{y}) \leq \mathbf{0}$. The solution is obtained with a Lagrange multipliers approach. The minimization of the Lagrangian

$$\boldsymbol{L}(\boldsymbol{y}, \boldsymbol{\lambda}) = F(\boldsymbol{y}) - \tilde{\boldsymbol{\lambda}}^T \boldsymbol{c}(\boldsymbol{y}), \tag{13}$$

where $\tilde{\boldsymbol{\lambda}}^T$ corresponds to the Lagrange multipliers, leads to a set of Karush-Kuhn-Tucker (KKT) conditions, whose solution yields the optimal control history and corresponding trajectory. Therefore, the pair $(\boldsymbol{x}^*, \tilde{\boldsymbol{\lambda}}^*)$ is an optimum if it satisfies the necessary conditions [2]

$$\begin{cases} \nabla_{\mathbf{y}} L = \mathbf{g}(\mathbf{y}) - \mathbf{G}^\top(\mathbf{y})\tilde{\boldsymbol{\lambda}} = \mathbf{0} \\ \nabla_{\lambda} L = -\mathbf{c}(\mathbf{y}) = 0 \\ \tilde{\lambda}_i^* \geq 0 \qquad \text{for all } i = 1, ..., n_A \end{cases} \tag{14}$$

where ∇ indicates the derivatives the Lagrangian with respect to $\boldsymbol{y}$ and $\boldsymbol{\lambda}$, and i stands for the n_A active constraints. The system (14) can be solved using a Newton method, and $\boldsymbol{g}$ and $\boldsymbol{G}$ in the formulas are respectively the objective function gradient and the Jacobian matrix of the constraints, mathematically defined as

$$\mathbf{g}(\mathbf{y}) = \left(\frac{\partial F}{\partial \mathbf{y}}\right)^\top = \begin{bmatrix} \frac{\partial F}{\partial y_1} \\ \vdots \\ \frac{\partial F}{\partial y_n} \end{bmatrix}, \qquad \mathbf{G}(\mathbf{y}) = \frac{\partial \mathbf{c}(\mathbf{y})}{\partial \mathbf{y}} = \begin{bmatrix} \frac{\partial \mathbf{c_1}}{\partial y_1} & \dots & \frac{\partial \mathbf{c_1}}{\partial y_n} \\ \vdots & & \\ \frac{\mathbf{c_m}}{\partial y_1} & \dots & \frac{\mathbf{c_m}}{\partial y_n} \end{bmatrix}. \tag{15}$$

Comparison

Now that the methods have been described, their main features can be summarized. By considering the procedure adopted, indirect methods are often referred to as *optimize then discretize* [2], since the first-order necessary conditions for optimality are analytically derived and solved [14]. Their primary advantage is the high accuracy of the solution, but the necessary conditions need to be re-computed every time the problem statement changes. Furthermore, the problem is very sensitive to the initial guess of the costates, which is not intuitive, causing poor convergence properties and lack of robustness [15].

On the other hand, direct methods transcribe the original continuous-time OCP into a NLP. Therefore, they *discretize then optimize*. They have broader convergence properties and thus are easier to initialize, but the lack of first-order necessary optimality conditions derivation yields to uncertainties in the nature of the solution. Path constraints are easily handled, making these techniques appealing for large and complex optimization problems. However, a large number of variables is generally involved, leading to numerical issues and requiring high computational effort [16].
The main properties are summarized in Table 1.

Table 1 Direct and indirect methods main features.

Method	Flexibility	Robustness	Optimality
Indirect	×	×	✓
Direct	✓	✓	×

3 Covector Mapping Principle

After an overview of numerical methods has been given, we present the Covector Mapping Principle (CMP). As already discussed, low-thrust trajectory optimization problems are treated with either direct or indirect methods. Promising techniques are continuously developed and updated thanks to progress in computer technologies and mathematics [3], but they always treat the two categories of methods as

separate. However, they originate from the same continuous-time OCP, and therefore the existence of a link between them seems intuitive. In the last decades, researchers came up with the idea that Lagrange multipliers of the direct approach are actually a discretized version of the costate variables resulting from the indirect formulation [17–19]. The aim of the CMP is to exploit this relation and build an algorithm that combines convergence properties of direct methods and accuracy of indirect ones. First of all, the concept behind this method is contextualized in Optimal Control Theory by means of functional analysis, and in particular by the *Hahn-Banach theorem*. The reader should refer to [20] for a detailed study. Note that the notation in this section differs from the rest of the paper. If a generic vector Ordinary Differential Equation (ODE) and a functional $\mathcal{H}$ are such that

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}), \tag{16}$$

$$\mathcal{H}(\boldsymbol{\gamma}, \mathbf{x}) := \boldsymbol{\gamma}^T \mathbf{f}(\mathbf{x}), \tag{17}$$

where $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ is continuous and differentiable, $\boldsymbol{x}, \boldsymbol{\gamma} \in \mathbb{R}^n$, and $\boldsymbol{\gamma}$ is an auxiliary vector of variables. Then, an auxiliary ODE can be defined as

$$-\dot{\boldsymbol{\gamma}} := \left(\frac{\partial \mathcal{H}}{\partial \mathbf{x}}\right)^T = \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^T \boldsymbol{\gamma}. \tag{18}$$

Since it is always possible to define $\mathcal{H}$, it follows that the auxiliary ODE always exists because Equation (16) exists. Therefore, Equation (16) can always be transformed to a higher-dimensional space, the *primal-dual space* [4]. This dualization process allows to reinterpret direct and indirect methods as part of the same formulation. Indeed, the definition of necessary optimality conditions in the context of indirect methods, Equation (7), is the result of the dualization through the Legendre-Fenchel transform [21], while a direct method can instead be interpreted as the solution of only half of the Hamiltonian system, thus ignoring the optimality of the solution. As a consequence, the CMP exploits the hidden Hamiltonian structure to reach its goal.

3.1 OCP formulations

To better understand the method underlying logic, different formulations of the same OCP, denoted by P, are introduced. They are represented in Figure 1. Problem $\boldsymbol{P}$ is the original continuous-time OCP. As shown in section 2, it can be dualized, obtaining the TPBVP $\boldsymbol{P}^{\lambda}$. It is then solved through numerical integration, therefore introducing discretization. This formulation corresponds to $\boldsymbol{P}^{\lambda N}$. The path $\boldsymbol{P} \to \boldsymbol{P}^{\lambda} \to \boldsymbol{P}^{\lambda N}$ in orange in Figure 1 is the variant solved by indirect methods. On the other hand, $\boldsymbol{P}$ can also be firstly discretized, as in direct approaches, becoming $\boldsymbol{P}^{N}$, and a zero-finding version $\boldsymbol{P}^{N\lambda}$ of $\boldsymbol{P}^{N}$ is obtained through dualization. This is the most implemented version in commercial software [22]. The CMP follows the *hybrid* technique in blue in Figure 1. In particular, problem $\boldsymbol{P}$ is firstly discretized using a method that satisfies the CMP, resulting in $\boldsymbol{P}^{N}$. It is subsequently dualized becoming $\boldsymbol{P}^{N\lambda}$, and solved. The CMT is exploited to compute the set of costate variables, which are then used to verify the necessary optimality conditions [3]. Therefore, the direct method is used as initial guess generation mechanism for an indirect method, and this technique allows to avoid the issues caused by the use of an indirect approach only, being both accurate and robust.

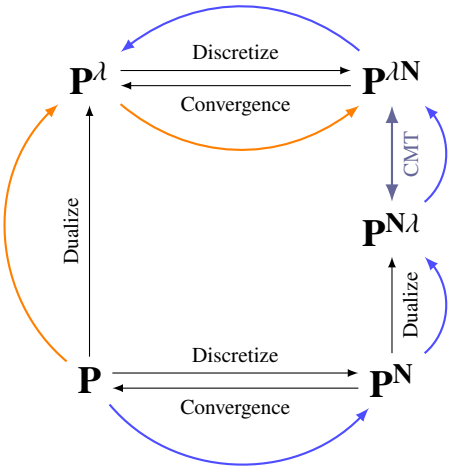

Fig. 1 OCP formulations. In blue the *hybrid method*. In orange the *indirect strategy*.

The CMP is defined as follows: given both the solution to the continuous OCP denoted by $\boldsymbol{P}$ and a series of approximations of $\boldsymbol{P}$ converging to it, the CMP states that, under certain circumstances dictated by the CMT, there exist a sequence of solutions of the approximated problem leading to the solution of the original one.

Dual variables are the central characters of the CMP, and they correspond to costate variables in indirect methods and to Lagrange multipliers in the direct approach. More specifically, the mapping consists in finding the transformation of the multipliers between problems $\boldsymbol{P}^{N\lambda}$ and $\boldsymbol{P}^{\lambda N}$. Finding this map is not straightforward, since not all discretization methods allow its existence. Methods that provide a CMT are referred to as *complete* methods [4], and they include the Symplectic Euler method [23], which however is characterized by poor accuracy, Hager's family of Runge-Kutta methods [3], that are as accurate as the order of the method employed, and PS methods [6]. The last ones are the most promising, being characterized by a very fast convergence rate, known as spectral accuracy, and by an Eulerian-like simplicity [24].

3.2 CMP with Non-Complete methods

The crucial step to reach the goal of this work is to provide a mapping between Lagrange multipliers and costates when a non-complete method is used in order to verify if and to what extent the CMP can be exploited. Since direct and indirect methods have different formulations, these are recalled. The expression of the augmented cost functional of the indirect approach and the Lagrangian involved in the direct method are respectively

$$\hat{J} = J_{\mathrm{EO}}^{L} + \boldsymbol{\nu}^{T}\boldsymbol{\phi}(\boldsymbol{x}(t_0), t_0, \boldsymbol{x}(t_f), t_f) + \int_{t_0}^{t_f} \boldsymbol{\lambda}^{T}(t)(\boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) - \dot{\boldsymbol{x}}(t))dt, \tag{19}$$

$$\boldsymbol{L}(\boldsymbol{y}, \boldsymbol{\lambda}) = J_{\mathrm{EO}}^{L} - \tilde{\boldsymbol{\lambda}}^{T}\boldsymbol{c}(\boldsymbol{y}), \tag{20}$$

where J_{EO}^{L} is the Lagrange cost functional of the EO problem, defined in Equation (2). Note that the boundary conditions introduced by $\boldsymbol{\nu}$ in $\hat{J}$ are embedded in the vector of constraints $\boldsymbol{c}(\boldsymbol{y})$ in $\boldsymbol{L}$. Therefore, by comparing the two equations, it is clear that

$$\tilde{\boldsymbol{\lambda}}_i \approx -\boldsymbol{\lambda}(t_i), \qquad \text{for all } i = 1, ..., N-1, \tag{21}$$

where t_i is the discrete time instant. The last node corresponding to $i = N$ is not included using Hermite-Simpson collocation because defect constraints are imposed at grid mid-points.

This approximation is considered acceptable, although it seems less accurate than results reported in literature [2, 17]. Indeed, known relations assume that both direct and indirect methods employ the same numerical scheme, and this does not apply within this work. In spite of this, Equation (21) represents a good initial guess for the optimization process.

4 Simulation and Results

In this section the EO low-thrust trajectory optimization problem is solved using different algorithms to show to what extent the CMP can be exploited with non-complete methods. In particular, Hermite-Simpson direct collocation scheme is adopted. A comparison is performed by solving the same problem with a complete global PS methods employing LGR collocation. To perform the mapping, Equation (21) and the CMT provided in [6] are used in the non-complete and the complete frameworks, respectively. Concerning the indirect formulation, a RK7-8 single-shooting method is used [8, 25]. The most relevant settings of the problem are summarized in section 4.1, while the results and analysis concerning non-complete and complete steps are analyzed in sections 4.2, 4.3 and 4.4.

4.1 Simulation settings

The problem is formulated in Lagrange form and in Cartesian coordinates within each solver in order to reduce the source of error concerning transformations when comparing the results, and also when the solution obtained with a method is used as warm start for another. Note, however, that the optimization tools present a substantial difference, as they employ different discretization methods. This feature influences the application of the CMP, and therefore it has to be considered in the analysis of the solution. It would have had more importance if the aim of the simulation was to provide a CMT, since the same discretization shall be used to close the gap between the dual variables [4].

The solution of the root-finding problem involved in the indirect formulation relies on $MATLAB^{®}$ `fsolve`, while the direct algorithm is implemented in the framework offered by CasADi tool [26], adopting interior-point method. The results are relative to *AMD Ryzen 9 5950X 16-Core Processor, 3.40 GHz, RAM 128 GB*. The problem is solved using initial and final conditions reported in Table 2, where a fixed Time of Flight (ToF) is imposed. All the variables are scaled to non-dimensional units prior to the optimization process for numerical reasons. This procedure is common to all solvers, and the scaling factors are reported in Table 3, together with the values of physical constants involved in the problem formulation. Note that the values related to the engine, $T_{\max}$ and I_{sp}, refer to M-ARGO mission [27]. Concerning instead the number of collocation points, 70 nodes are used for the non-complete method, as they represent a good compromise between accuracy and computational time. This value can be increased without any issue for the solver. On the other hand, a large number of points in the complete solution would cause ill-conditioning of Legendre polynomials. Therefore, the accuracy of the results is sensitive to the number of collocation points, as visible in Figure 8. After precision evaluation of the solution, 23 nodes are chosen.

Table 2 Boundary conditions of the analyzed problem.

Parameter	Value	Unit
$\boldsymbol{r_0}$	[1.51·10^8; -2.24·10^7; 4.49·10^6]	km
$\boldsymbol{v_0}$	[5.36; 30.1; 0]	km s^{-1}
m_0	23	kg
$\boldsymbol{r}_f$	[1.54·10^8; -2.09·10^7; 4.49·10^6]	km
$\boldsymbol{v}_f$	[3.92; 28.93; 0.84]	km s^{-1}
ToF	420	days

Table 3 Scaling factors and physical constants.

Parameter	Value	Unit
Length Unit (LU)	1.49598·10^8	km
Time Unit (TU)	3.15576·10^7	s
Velocity Unit (LU/TU)	4.7405	km s^{-1}
Mass Unit (MU)	10	kg
$\mu_\odot$	1.32712·10^{11}	km^3 s^{-2}
g_0	9.80665	m s^{-2}
$T_{\max}$	2.3·10^{-3}	N
I_{sp}	3000	s

4.2 Non-complete step

First of all, the EO problem is solved with the indirect IND and the direct Hermite-Simpson HS methods independently, as visible in the graph in Figure 2, in order to obtain the optimal solutions in terms of dual variables. They are used to compute the mapping from costates to Lagrange multipliers and viceversa as in section 3.2. This is the most important step of the analysis, as identifying a correct transformation will allow to exploit the CMP. Optimized solutions in Figure 3 show that the shape of dual variables is essentially the same, and an almost exact superposition is observable. However, a scaling factor equal to approximately 4.1 between Lagrange multipliers $\tilde{\boldsymbol{\lambda}}$ and costates $\boldsymbol{\lambda}$ has been applied, namely $k := \tilde{\boldsymbol{\lambda}}/\boldsymbol{\lambda} \approx 4.1$. This is still an open issue, and a brief analysis of possible causes is presented in section 4.4.

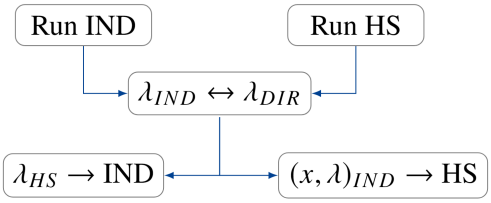

Fig. 2 Non-complete step procedure.

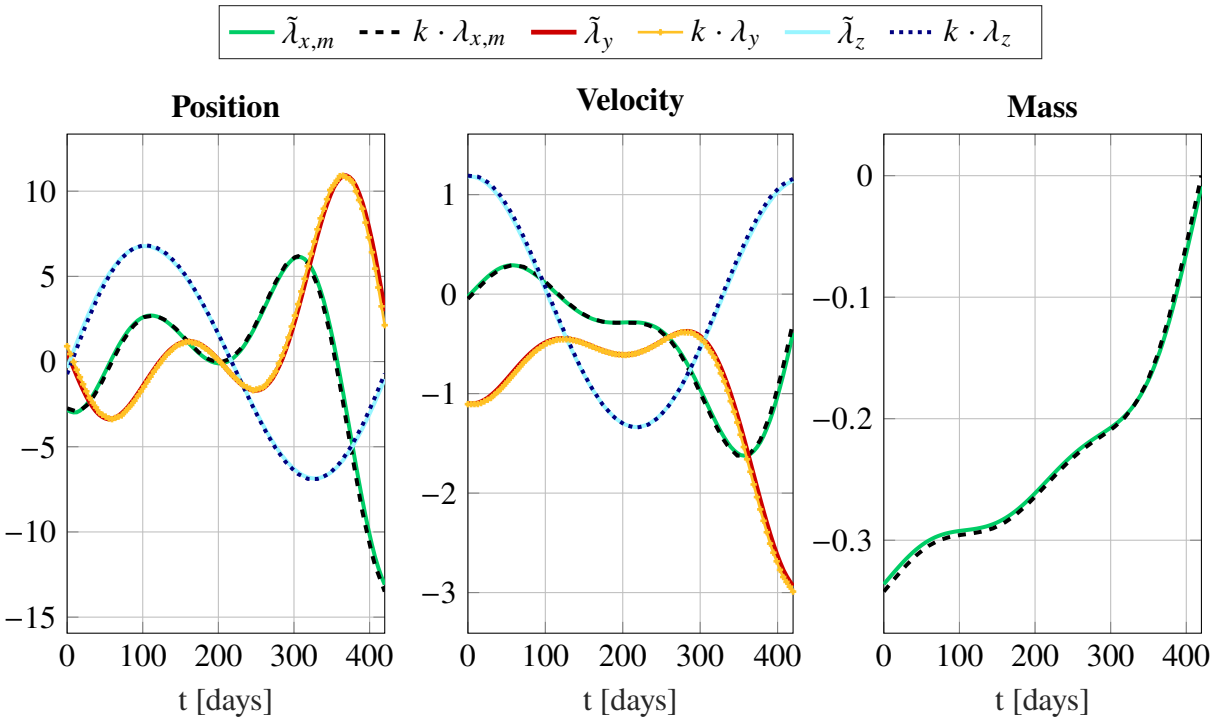

Fig. 3 Lagrange multipliers $\tilde{\lambda}$ of non-complete method and costates λ multiplied by $k = 4.1$.

To analyze the performance two simulations are leveraged.

1) The initial costates obtained from the mapping of Lagrange multipliers from the direct solution are used as initial guess within the indirect solver both with scaling factor, $\boldsymbol{\lambda_{HS_{\text{mapped}}}}$, and without it, $\boldsymbol{\lambda_{HS}}$, to verify whether the indirect solver still identifies a mathematical connection. The results in Table 4 show the value of the shooting function $\boldsymbol{f}(\boldsymbol{x})$ at different simulation iterations and the computational time. Number of evaluated steps is the same in both cases, and also CPU time is similar and limited. Therefore the mapping works and can be used even without scaling factor. Note that the CPU time throughout the work refers to the average computational time over 10 simulations.
2) The dual variables obtained from the indirect algorithm are mapped according to Equation (21) and given as input to CasADi by allowing the warm start option of the solver. Table 5 shows the results related to the cold start case in the first column, denoted with HS, while those obtained with the warm start are reported in the second and third column. In particular, $\boldsymbol{x}_{IND \rightarrow HS}$ are

the results obtained with initialization of states only, while $(\boldsymbol{x}, \boldsymbol{\lambda})_{IND \to HS}$ are those obtained with initialization of both states and dual variables. They show a marked improvement in the computational time and number of iterations required compared to the cold start solution. In addition to this, the warm start that includes also the dual variables is even more beneficial for the solver. It can be concluded that this analysis proves the mapping to be successful also in the passage from indirect to direct methods in terms of computational cost.

Table 4 Direct non-complete $\tilde{\lambda}_{HS} \to$ Indirect solver.

	$\boldsymbol{\lambda_{HS}}$	$\boldsymbol{\lambda_{HS_{\text{mapped}}}}$
Step	$0 \to 14$	$0 \to 14$
$\mathbf{f(x)}$	$9.88 \to 1.58 \cdot 10^{-20}$	$1.19 \to 3.97 \cdot 10^{-27}$
CPU Time	0.592 s	0.511 s

Table 5 Indirect solution $(\boldsymbol{x}, \boldsymbol{\lambda})_{IND} \to$ Direct non-complete solver.

	HS	$\boldsymbol{x}_{IND \to HS}$	$(\boldsymbol{x}, \boldsymbol{\lambda})_{IND \to HS}$
# Iterations	59	29	22
CPU Time	1.139 s	0.549 s	0.422 s

Remark: a consistent number of steps is still required to reach the solution in warm start cases. This is mainly due to the difference in the discretization schemes adopted within the optimization tools. It is also important to highlight that solutions converge to cold start ones in Figures 5 and 6. Therefore, the mapping from indirect to direct method lowers the computational cost, but does not improve optimality of the results.

4.3 Complete step

The same problem is now solved using PS method with LGR collocation. As already mentioned, even in this case the discretization method is different from the one used in the indirect approach. However, since the CMT [6] provides the link between dual variables, and according to [28] it is not necessary to develop the indirect PS collocation method to obtain the performances of an indirect method while employing a direct one, it is expected that the limit posed by the discretization is at least partially overcome. The procedure followed for the analysis is the same performed in the previous section, and it is shown in Figure 4. The optimized results obtained with PS algorithm are reported in Figures 5, 6 and 7, and show convergence to the optimal solution obtained with the indirect formulation. The Lagrange multipliers are more accurate compared to those obtained in the non-complete framework. However, the same scaling factor of 4.1 is found. The CMT is then applied to analyze the performances of the complete step. It is used to map the Lagrange multipliers into the costates and viceversa. If it works correctly, convergence to the optimal solution shall be reached faster and in fewer iterations with respect to the non-complete step simulations.

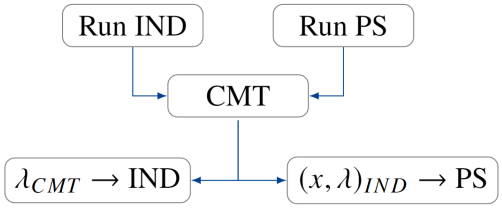

Fig. 4 Complete step procedure.

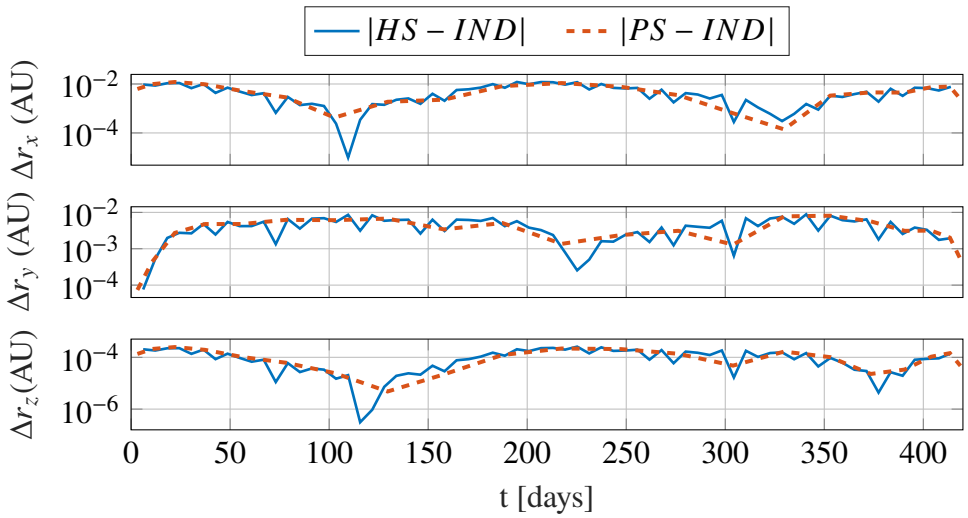

Fig. 5 Trajectory error with respect to indirect solution in logarithmic scale.

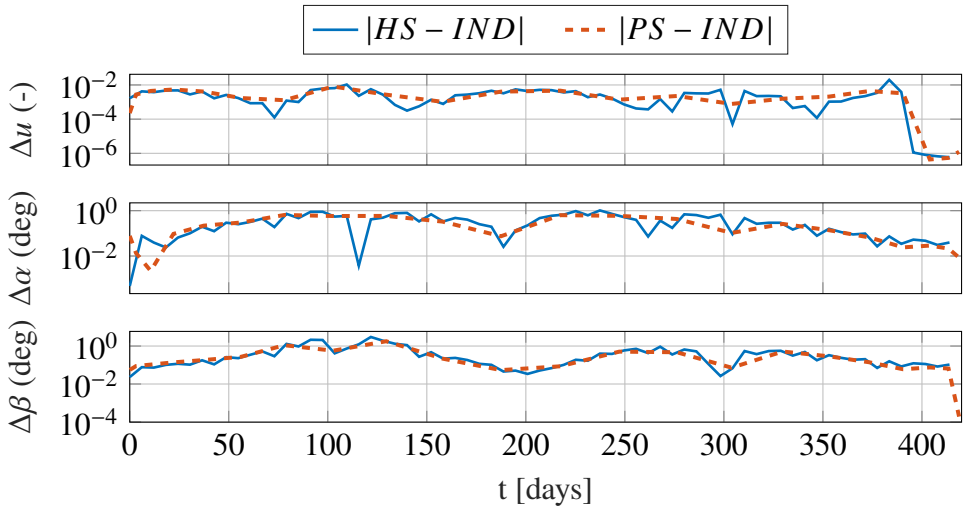

Fig. 6 Controls error with respect to indirect solution in logarithmic scale.

The warm start of the indirect solver with the Lagrange multipliers mapped through the CMT leads to the results in Table 6. The scaling factor of 4.1 has been applied in the simulation in the second column of the table, showing a great improvement both in the value of the shooting function $\boldsymbol{f}(\boldsymbol{x})$ and in the number of iterations needed to reach convergence. Compared to the results obtained in the section 4.2, the value of the shooting function is significantly lowered, due to the application of the analytical mapping offered by the CMT. This reflects on the computational time and the steps required to obtain the solution, thus showing the improvements given by the complete method.

Table 6 CMT → Indirect solver.

	λ_{CMT}	$\lambda_{CMT_{\text{mapped}}}$
Step	$0 \rightarrow 7$	$0 \rightarrow 3$
f(x)	$0.717 \rightarrow 1.66{\cdot}10^{-19}$	$1.89^{-4} \rightarrow 1.98{\cdot}10^{-21}$
CPU Time	0.362 s	0.234 s

Also in the results obtained by warm starting the direct method shown in Table 7, it is observed that the already low computational time is greatly improved. However, the passage from indirect to direct solver

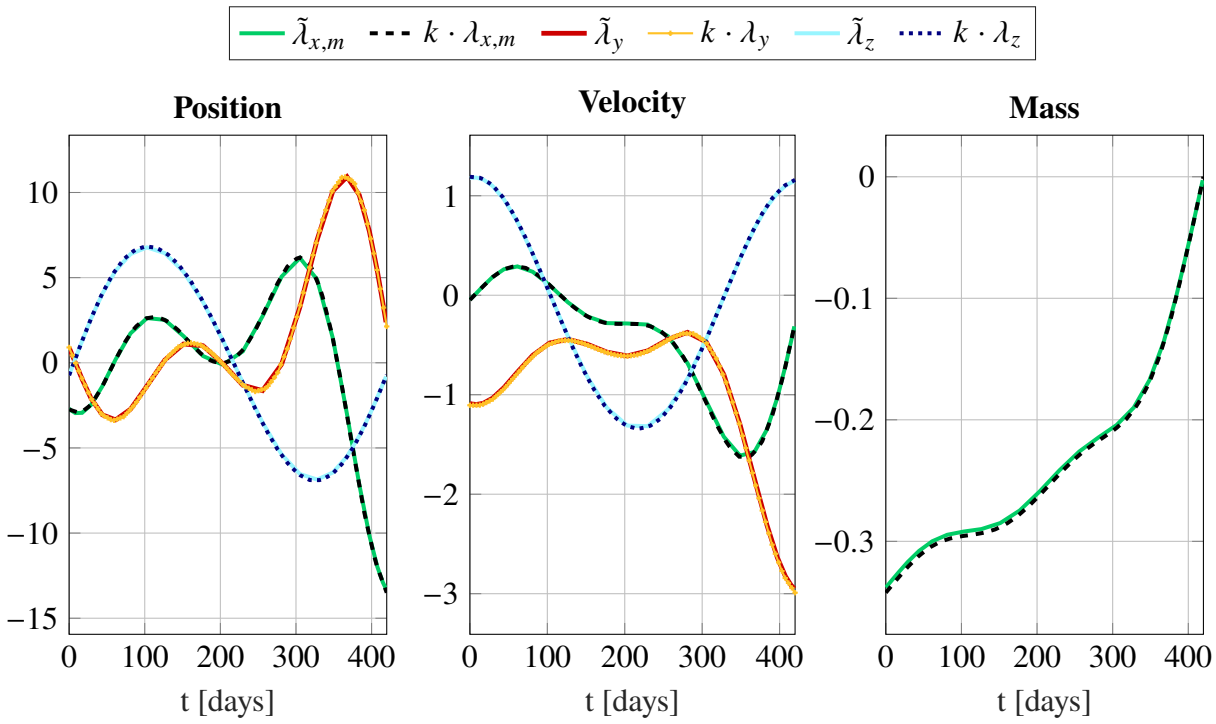

Fig. 7 Lagrange multipliers $\tilde{\lambda}$ of complete method and costates λ multiplied by $k = 4.1$.

is more sensitive to the difference in the discretization scheme, as a larger number of iterations is needed. The results are overall better compared to the non-complete solution.

Table 7 Indirect solution $(\boldsymbol{x}, \boldsymbol{\lambda})_{IND} \rightarrow$ Direct complete solver.

	PS	$\boldsymbol{x}_{IND \rightarrow PS}$	$(\boldsymbol{x}, \boldsymbol{\lambda})_{IND \rightarrow PS}$
# Iterations	136	11	15
CPU Time	0.321 s	0.074 s	0.076 s

Nonetheless, opposite to the non-complete case, the PS method is sensitive to the number of collocation points adopted, as visible in Figure 8, and therefore a significant user input is required to perform a trade-off between preventing polynomials ill-conditioning and obtaining good accuracy, thus employing a sufficient number of nodes. Still, even when a suboptimal solution obtained with 28 collocation points is used as warm start for the indirect solver, optimality with input $\boldsymbol{\lambda}_{CMT_{\text{mapped}}}$ is reached in only 5 iterations and CPU time of 0.267 s. However, this is not happening when going from indirect to direct method, thus requiring a good tuning of the number of nodes if the CMT is to be used in this case.

4.4 Scaling factor analysis

A scaling factor between the Lagrange multipliers and the costates has been observed in both the non-complete and the complete step. This is still an open issue, which may be related to the scaling factors that have been applied to the NLP variables. Indeed, efficiency and accuracy of the solution should be increased by the NLP variables scaling, but sometimes this procedure can induce an undesired effect on Lagrange multipliers and costates [29]. Therefore, even if the NLP variables become well-scaled, an opposite effect is obtained on the dual variables. The latter can become large, and their multiplication with the vector of constraints in the Lagrangian of the problem would lead to a poor, destabilized solution. This might cause either non-convergence of the solver or convergence to a pseudo-minimizer. The reader should refer to [30] for a deeper understanding. In the analyzed scenario, the factor of 4.1 does not prevent the solution to converge to the optimal one obtained with the indirect approach, as visible in Figures 5

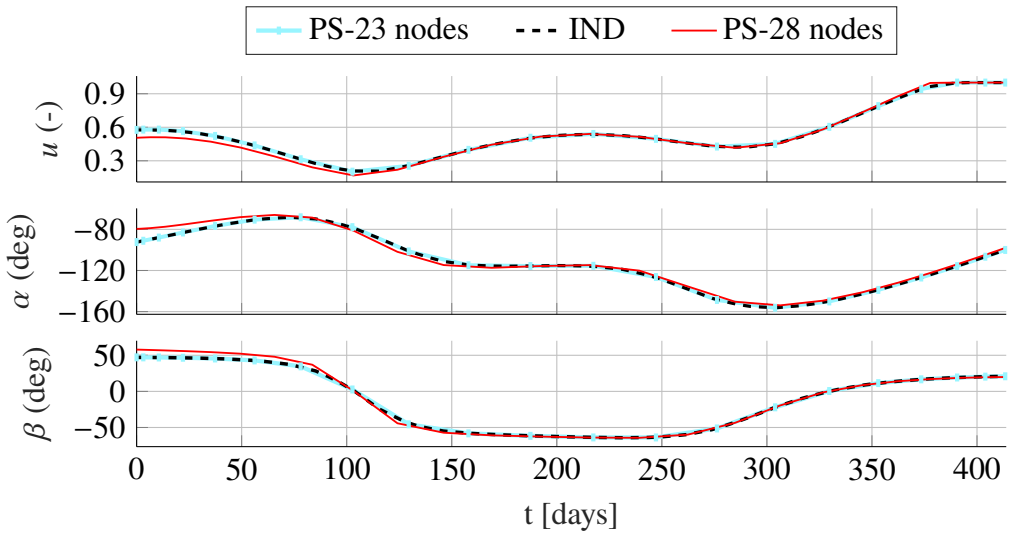

Fig. 8 Optimized controls at varying number of nodes for the PS method.

and 6, where the error is due to the nature of direct collocation itself. However, different approaches have been developed and will be used in future simulations to investigate the mentioned issue. Balancing techniques can be applied to manually tune the scaling in the direct optimization framework [29], but this would require significant user input. Therefore, automatic techniques as isoscaling [2], Jacobian rows normalization, and projected-Jacobian rows normalization [31] have been proposed. While the first one does not consider the relationship between states and constraints in the scaling procedure, the second only considers this correlation, ignoring the states normalization. The latter considers both aspects and also expands the method to dual variables scaling. Note that this further analysis would also allow to verify that the scaling issue is not due to a different variable scaling applied internally in the indirect solver.

5 Conclusion

The application of the Covector Mapping Principle to non-complete methods within the context of low-thrust trajectory optimization has been proven successful. It allows to close the gap between direct and indirect approaches when only solvers employing non-complete methods are available. Indeed, the empirical link between Lagrange multipliers and costate can be exploited passing from both direct to indirect methods and viceversa, allowing to lower the computational time and to overcome the intrinsic limitations of the two families of approaches. The comparison with the complete Pseudospectral method shows that better performances are obtained by applying the Covector Mapping Theorem, but since trajectory design is performed off-line, non-complete methods can be used with good accuracy of the solution and low computational cost. However, accuracy of the results is not improved when the direct method is warm started, therefore suggesting to use the CMP by first exploiting the larger convergence basin of direct techniques, and then improving the accuracy of the solution through an indirect method. Moreover, particular attention concerns the number of collocation points required by the Pseudospectral method in order to guarantee solution accuracy while preventing ill-conditioning of Legendre polynomials. Indeed, this issue would affect the application of the Covector Mapping Principle in the passage from indirect to direct approaches. Additional work is required to confirm the cause of the scaling factor between Lagrange multipliers and costates, since in different applications it may lead to poor accuracy solutions, or, in the worst case, to non-convergence of the solver. Moreover, the same discretization scheme should be employed in all solvers to check if the performances would improve.

References

[1] G.P. Sutton and O. Biblarz. *Rocket Propulsion Elements*. A Wiley Interscience publication. Wiley, 2001. ISBN: 9780471326427.

[2] John T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming, Second Edition*. Society for Industrial and Applied Mathematics, second edition, 2010. DOI: 10.1137/1.9780898718577.

[3] I. Michael Ross. A Historical Introduction to the Covector Mapping Principle. 2005. `https://api.semanticscholar.org/CorpusID:15162480`.

[4] I. Michael Ross and F. Fahroo. *A Perspective on Methods for Trajectory Optimization*. DOI: 10.2514/6.2002-4727.

[5] Hans Seywald and Renjith R. Kumar. Method for automatic costate calculation. *Journal of Guidance, Control, and Dynamics*, 19(6):1252–1261, 1996. DOI: 10.2514/3.21780.

[6] D. Garg. *Advances in global pseudospectral methods for optimal control*. PhD thesis, University of Florida, 08 2011.

[7] Fabio Spada, Marco Sagliano, and Francesco Topputo. Direct–indirect hybrid strategy for optimal powered descent and landing. *Journal of Spacecraft and Rockets*, 60(6):1787–1804, 2023. DOI: 10.2514/1.A35650.

[8] Yang Wang. *Efficient Indirect Optimization of Low-Thrust Trajectories with Interior-Point Constraints*. PhD thesis, Politecnico di Milano, 2 2022.

[9] James M. Longuski, José J. Guzmán, and John E. Prussing. *Optimal Control with Aerospace Applications*. Springer, 2014.

[10] A. E. Bryson and Y. Ho. *Applied Optimal Control: Optimization, Estimation and Control*. Routledge, 1975. DOI: 10.1201/9781315137667.

[11] L. S. Pontryagin. *The Mathematical Theory of Optimal Processes*. CRC Press, 1987.

[12] F. Topputo and C. Zhang. Survey of Direct Transcription for Low-Thrust Space Trajectory Optimization with Applications. *Abstract and Applied Analysis*, 2014:1–15, 2014. DOI: 10.1155/2014/851720.

[13] Albert L. Herman and Bruce A. Conway. Direct optimization using collocation based on high-order gauss-lobatto quadrature rules. *Journal of Guidance, Control, and Dynamics*, 19(3):592–599, 1996. DOI: 10.2514/3.21662.

[14] John T. Betts. Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 21(2):193–207, 1998. DOI: 10.2514/2.4231.

[15] David Morante, Manuel Sanjurjo Rivo, and Manuel Soler. A survey on low-thrust trajectory optimization approaches. *Aerospace*, 8(3), 2021. ISSN: 2226-4310. DOI: 10.3390/aerospace8030088.

[16] Bruce A. Conway. *Spacecraft Trajectory Optimization*. Cambridge Aerospace Series. Cambridge University Press, 2010. DOI: 10.1017/CBO9780511778025.

[17] O. von Stryk and R. Bulirsch. Direct and indirect methods for trajectory optimization. *Annals of Operations Research*, 37(1):357–373, 1992. DOI: 10.1007/BF02071065.

[18] Oskar von Stryk. *Numerical Solution of Optimal Control Problems by Direct Collocation*, pages 129–143. Birkhäuser Basel, Basel, 1993. ISBN: 978-3-0348-7539-4. DOI: 10.1007/978-3-0348-7539-4_10.

[19] Paul J. Enright and Bruce A. Conway. Discrete approximations to optimal trajectories using direct transcription and nonlinear programming. *Journal of Guidance, Control, and Dynamics*, 15(4):994–1002, 1992. DOI: 10.2514/3.20934.

[20] T. Tao. *An Epsilon of Room Real Analysis: Pages from Year Three of a Mathematical Blog*. Graduate Studies in Mathematics. American Mathematical Society, 2011.

[21] I. M. Ross. An optimal control theory for nonlinear optimization. *Journal of Computational and Applied Mathematics*, 354:39–51, 2019. ISSN: 0377-0427. DOI: https://doi.org/10.1016/j.cam.2018.12.044.

[22] Michael A. Patterson and Anil V. Rao. Gpops-ii: A matlab software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming. *ACM Trans. Math. Softw.*, 41(1), oct 2014. ISSN: 0098-3500. DOI: 10.1145/2558904.

[23] I. Michael Ross. A roadmap for optimal control: The right way to commute. *Annals of the New York Academy of Sciences*, 1065(1):210–231, 2005. DOI: 10.1196/annals.1370.015.

[24] Wei Kang, Qi Gong, and I.M. Ross. Convergence of pseudospectral methods for a class of discontinuous optimal control. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 2799–2804, 2005. DOI: 10.1109/CDC.2005.1582587.

[25] F. Topputo, D. A. Dei Tos, K. V. Mani, S. Ceccherini, C. Giordano, V. Franzese, and Y. Wang. Trajectory Design in High-Fidelity Models. In *7th International Conference on Astrodynamics Tools and Techniques (ICATT)*, pages 1–9, Oberpfaffenhofen, Germany, 11 2018. `https://hdl.handle.net/11311/1068813`.

[26] Joel A. E. Andersson, Joris Gillis, Greg Horn, James B. Rawlings, and Moritz Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019. DOI: 10.1007/s12532-018-0139-4.

[27] Francesco Topputo, Yang Wang, Carmine Giordano, Vittorio Franzese, Hannah Goldberg, Franco Perez-Lissi, and Roger Walker. Envelop of reachable asteroids by M-ARGO CubeSat. *Advances in Space Research*, 67(12):4193–4221, 2021. ISSN: 0273-1177. DOI: 10.1016/j.asr.2021.02.031.

[28] I. Michael Ross and Fariba Fahroo. Legendre Pseudospectral Approximations of Optimal Control Problems. In Wei Kang, Carlos Borges, and Mingqing Xiao, editors, *New Trends in Nonlinear Dynamics and Control and their Applications*, pages 327–342, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. ISBN: 978-3-540-45056-6. DOI: 10.1007/978-3-540-45056-6_21.

[29] I. M. Ross, Q. Gong, M. Karpenko, and R. J. Proulx. Scaling and balancing for high-performance computation of optimal controls. *Journal of Guidance, Control, and Dynamics*, 41(10):2086–2097, 2018. DOI: 10.2514/1.G003382.

[30] Samule Sowell. The Tiger Optimization Software - A Pseudospectral Optimal Control Package. Master's thesis, Auburn University, Auburn, Alabama, December 2022.

[31] Marco Sagliano, Stephan Theil, Michiel Bergsma, Vincenzo D'Onofrio, Lisa Whittle, and Giulia Viavattene. On the Radau pseudospectral method: theoretical and implementation advances. *CEAS Space Journal*, 9(3):313–331, Sept. 2017. DOI: 10.1007/s12567-017-0165-5.