

Article

Ascent Trajectory Optimization Using Second-Order Birkhoff Pseudospectral Methods

Xiaopeng Xue ^{1,2} , Yujia Xie ^{3,4}, Hui Zhou ^{3,4,*}, Qinghai Gong ^{3,4} and Dangjun Zhao ^{1,2} 

¹ School of Automation, Central South University, Changsha 410083, China; xuexiaopeng@csu.edu.cn (X.X.); zhao_dj@csu.edu.cn (D.Z.)

² Hunan Provincial Key Laboratory of Optic-Electronic Intelligent Measurement and Control, Changsha 410083, China

³ Beijing Aerospace Automatic Control Institute, Beijing 100854, China; xyj1867@outlook.com (Y.X.); gqh_calt12@126.com (Q.G.)

⁴ National Key Laboratory of Science and Technology on Aerospace Intelligent Control, Beijing 100854, China

* Correspondence: zhouh_12@163.com

Abstract: This paper proposes a novel convex optimization framework for two-stage launch vehicle (TSLV) ascent trajectory planning from liftoff to orbit insertion, featuring two ground-breaking advancements over existing convex optimization methodologies: (1) an innovative second-order Birkhoff pseudospectral (BPS) method is developed that reduces the number of dynamic equality constraints by 50% compared to traditional PS methods, meanwhile, an augmented variable transcription technique is used to formulate inequality constraints; therefore, the sparsity ratio of the inequality matrix is reduced to less than 1%; (2) a new iterative solution strategy initialized by a few guesses is proposed to efficiently obtain the optimal solution. The framework is rigorously supported by theoretical convergence guarantees and validated through comprehensive numerical experiments. The numerical results demonstrate around a 50% reduction in computational time compared to the differential PS baseline method. With the significantly reduced computational cost, the proposed method exhibits strong potential for real-time onboard implementation in the future.

Keywords: ascent trajectory optimization; second-order Birkhoff pseudospectral method; two-stage launch vehicle; sequential convex optimization



Academic Editor: Vladimir S. Aslanov

Received: 7 January 2025

Revised: 6 February 2025

Accepted: 12 February 2025

Published: 13 February 2025

Citation: Xue, X.; Xie, Y.; Zhou, H.; Gong, Q.; Zhao, D. Ascent Trajectory Optimization Using Second-Order Birkhoff Pseudospectral Methods. *Aerospace* **2025**, *12*, 141.

<https://doi.org/10.3390/aerospace12020141>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

At present, launch vehicles (LVs) are the main tool for access to space. The payload is only a small fraction of the LV's initial mass, hence the expensive launch cost. The ascent trajectory optimization, a complex nonlinear optimal control problem (OCP) in nature, can provide a feasible way to increase the launcher capacity and reduce the overall launch cost. However, it is difficult to solve the ascent trajectory optimization problem with strong nonlinear dynamics as well as complex constraints. Historically, the offline-designed trajectory using the gravity turn technique is often used to open-loop guide the endo-atmospheric flight, while the closed-loop guidance strategies for the exo-atmospheric flight have been well developed, resulting in a non-optimal ascent flight and significant reduction in the payload mass [1]. In recent years, with the rapid development of computer science and space technologies, the need for reliable and efficient optimization methods is not only in the preliminary design phase but also in the optimization-based real-time guidance algorithms, where computational efficiency and robustness are primary considerations [2].

In general, OCPs can be solved by direct and indirect methods. As for the ascent exo-atmospheric trajectory optimization problem, indirect methods, usually based on

Pontryagin's Maximum Principle, have been used for decades, mainly resulting in near-optimal guidance algorithms, such as the iterative explicit guidance algorithms, and the closed-loop guidance strategies for LV [1,3–5]. Among these algorithms, many special techniques should be used to simplify the resulting Hamiltonian boundary value problem, therefore capable of real-time implementations [6]. The direct methods transcribe an OCP into a finite-dimensional nonlinear programming (NLP) problem by discretizing the states and the controls at a set of mesh points, which should be carefully designed to compromise the computational burden and the solution's accuracy. In general, the larger the number of mesh points, the more accurate the solution, as well as the larger the computational cost, including CPU time and memory [7]. Many numerical methods have been developed for the transcription of OCPs, and the most common method is by using a Pseudospectral (PS) collocation scheme, which is an optimal choice of mesh points in the reason of well-established rules of approximation theory [8]. There exist several mature optimal control toolkits based on the PS methods, such as DIDO and GPOPS [9,10]. The resulting NLP problems can be solved by well-known algorithm packages, such as IPOPT or SNOPT [11,12]. However, these NLP algorithms are sensitive to the choice of initial guess, and the obtaining of a global optimal solution cannot be necessarily guaranteed for complex NLP problems with multiple local minima; furthermore, these methods cannot provide a priori guarantees on the convergent rate [13].

In recent years, because of the polynomial-time complexity and the unique theoretical advantages, convex optimization methods have been widely used for rapidly solving complex ascent trajectory optimization problems [14–17]. The convex programming for trajectory optimization originates from the reference [13], in which the original OCP is often converted into a second-order conic programming (SOCP) problem by using the lossless convexification technique proposed in and discretization on uniformly distributed points. For complex nonlinear ascent trajectory optimization problems, the sequential linearizing techniques are used for the nonlinear dynamics constraints along the given trajectory; therefore, a series of SOCP problems can be iteratively solved through a sequential convex programming scheme. In general, a large number of discretizing points should be chosen for necessary accuracy, and this leads to extra computational burden [18]. To further accelerate the solving speed, therefore achieving the goal of online implementation, researchers have made a lot of effort. One way is to improve the solving procedure as well as the convex solver itself. In the study of [19], a warm-start of the interior-point method is applied to the sequential convex programming, and a GPU-based parallel preconditioned conjugate gradient method is reported in the study of [20]. Another way is to reduce the scale of the resulting SOCP problems by using PS methods for the transcription of the dynamic constraints in OCPs [2,21–24]. Due to the unique collocation strategy and global approximation characteristics, PS methods can achieve the same accuracy as uniform discretization with fewer collocation points.

In conventional PS methods, the differential matrix resulting from different collocation methods is used for the transcription of the dynamic equations, and this may lead to ill-conditioned SOCP problems when increasing the number of collocation points for a better accuracy of the solution [25]. Unfortunately, the ill-conditioned problem will arouse numerical instabilities and undermine the computational efficiency of the solver. To overcome the ill-conditioned problems in PS methods, *hp* schemes are proposed to divide the total collocation interval into several segments, and then, pseudospectral discretization is implemented in each segment [21,22]. Recently, by combining the PS collocation method and the Chebyshev Picard iteration, a more efficient semidefinite convex programming (SCP) algorithm is proposed in the study of [26], in which the OCP is transcribed into a sequence of finite-dimensional SCP subproblems suiting parallel implementations for

further speedup. From the perspective of reducing the scale and the condition number of the resulted SOCP problems for the typical cascaded integral systems, the authors of [27] proposed a Birkhoff PS (BPS) method based on the second-order Birkhoff interpolation polynomials to formulate well-conditioned SOCP problems even with a large number of collocation points. In the second-order BPS case, the number of decision variables as well as equality constraints are significantly reduced, therefore enhancing the computational efficiency. The proposed BPS methods are used for the rescue orbit search of launch vehicles in the study of [27] and the proximity operation of spacecraft in [28]. The results show that the BPS method can improve the numerical stability and the computational efficiency.

It is to be noted that only the vacuum flight motions are considered, and the complex path constraints and free-final time problems are not considered in the works [27,28]; consequently, the feasibility of the BPS method for solving the more general trajectory optimization problems with the strong nonlinear constraints still requires further research. Among launch vehicles, two-stage launch vehicles (TSLVs) present a unique set of challenges due to their complex dynamics during endo- and exo-atmospheric flight and the need for optimal ascent trajectories to achieve orbit insertion. In this paper, the ascent trajectory optimization problems constrained by the strong nonlinear dynamics of TSLVs and the maximum heat flux during the endo-atmospheric flight are considered for exploring the feasibility of the second-order BPS method in solving the complex multi-phase OCPs with free final time. Differing from the methodology reported by [15], in which the sequential linearization on the trajectory resulted from the last iteration solution for dealing with the dynamic constraints, the pseudo-linear system with state-depended coefficient matrices is used for the modeling of sequential convex optimization problems, therefore, avoiding the computational cost of linearization. Meanwhile, the trust region constraints imposed on the position and velocity vectors are not required in the proposed method, thus reducing the scale of the inequality constraints. The main contributions of this paper lie in two aspects: (1) proposing a new well-conditioned BPS transcription strategy for better sparsity patterns of the resulted SOCP problems, which will improve the algorithm's efficiency; and (2) proposing a sequential convex-solving procedure for the two-phase trajectory optimization problem with free-final time that can be initialized not by a three-dimensional reference trajectory, but a few guess values.

The remainder of this paper is organized as follows. Section 2 formulates the ascent trajectory optimization problem of a TSLV; then, the original problem is transcribed into its convexified version in Section 3. In Section 4, the sequential convex optimization problems are discretized via the well-conditioned Birkhoff PS method. The solution strategy and the numerical results, respectively, are presented in Sections 5 and 6. Finally, we conclude the paper in Section 7.

2. Problem Formulation

2.1. Ascent Dynamics

In this paper, a two-stage launch vehicle (TSLV) with liquid propellant rockets is considered, and its ascent flight consists of two phases: (1) the first phase is from launch firing to the stage separation, and (2) the second phase is from the separation to orbit insertion. For TSLVs, the stage separation height is around 60 km, and the atmosphere above 60 km is so rare that it can be approximated as a vacuum. Consequently, we assume that the first phase is endo-atmospheric flight, and the second phase is vacuum flight; therefore, the aerodynamics can be ignored. For the convenience of problem formulation, the following reasonable assumptions are made.

Assumption 1. *The thrust magnitude, as well as the mass flow rate of the rocket, are constant within a stage during the flight.*

Assumption 2. The Earth's gravity is modeled using a Newtonian central gravity field, with the assumption that the Earth is a sphere [1].

Remark 1. Assumption 1 simplifies the propulsion model by neglecting variations that occur in real-world scenarios. While this assumption provides a reasonable approximation for the majority of the ascent trajectory, it may introduce minor inaccuracies during critical phases such as stage separation or throttle adjustments. However, these deviations are typically small and can be mitigated through post-processing refinement or the use of a higher-fidelity propulsion model. Similarly, Assumption 2 neglects higher-order gravitational perturbations, which may become significant during high-altitude or orbital phases. For preliminary trajectory optimization, the spherical gravity model provides a computationally efficient approximation, with higher-fidelity models recommended for final verification or missions requiring high precision.

It is worth noting that the thrust magnitude of a liquid propellant rocket is dependent on the throttle setting. For a specific throttle value, the mass flow rate can be approximated by a constant [1]. Furthermore, the variation in the thrust magnitude resulting from changes in ambient atmospheric pressure is relatively so minor, allowing the thrust magnitude to be treated as nominally constant.

Based on the assumptions, the centroid motion of the TSLV is modeled in the Cartesian Earth-Centered Inertial (ECI) coordinate, and then the ascent flight dynamics including two phases are governed by the following model [17,29]:

$$\begin{bmatrix} \dot{\mathbf{R}}_i \\ \dot{\mathbf{V}}_i \\ \dot{m}_i \end{bmatrix} = \begin{bmatrix} \mathbf{V}_i \\ -\frac{\mu}{r_i^3} \mathbf{R}_i + \frac{\mathbf{D}_i}{m_i} + \frac{T_i}{m_i} \mathbf{u}_i \\ -q_i \end{bmatrix}, i = 1, 2, \quad (1)$$

where the subscript i represents the flight phase i , $\mathbf{R}_i = [x, y, z]^T \in \mathbb{R}^3$ and $\mathbf{V}_i = [v_x, v_y, v_z]^T \in \mathbb{R}^3$, respectively, are the position and velocity vectors of the launch vehicle in phase i , m_i is the mass of phase i , $r_i = \|\mathbf{R}_i\|$ is the geocentric radius of the vehicle, $\mu = 3.986012 \times 10^{14} \text{ m}^3/(\text{kg} \cdot \text{s}^2)$ is the gravitational constant of the Earth, T_i is the thrust magnitude of phase i , q_i is the mass flow rate of phase i , and $\mathbf{u}_i$ is the unit vector defining the thrust direction, i.e., the vehicle's body longitudinal axis. Similar to the studies of [14,29], for the TSLV with a slender rod structure, we ignore the aerodynamic lift force, and only the drag force $\mathbf{D}_i$ is considered as

$$\mathbf{D}_i = -\frac{1}{2} C_D S_{ref} \rho v_{i \text{ rel}} \mathbf{V}_{i \text{ rel}}, \quad (2)$$

where the coefficient of drag C_D is assumed to be a constant, S_{ref} is the vehicle's reference area, and the atmospheric density ρ depends on the current altitude $h = r - R_E$, and can be approximated by

$$\rho = \rho_0 e^{-h/H_s} \quad (3)$$

with $\rho_0 = 1.225 \text{ kg/m}^3$ and $H_s = 7200 \text{ m}$. $\mathbf{V}_{i \text{ rel}}$ is the relative velocity vector given by

$$\mathbf{V}_{i \text{ rel}} = \mathbf{V}_i - \boldsymbol{\omega}_E \times \mathbf{R}_i \quad (4)$$

where $\boldsymbol{\omega}_E$ is the angular rate vector of the earth. Correspondingly, $v_{i \text{ rel}}$ is the magnitude of $\mathbf{V}_{i \text{ rel}}$.

To facilitate the subsequent numerical calculations, it is necessary to normalize the variables in Equation (1): the distances are normalized by the earth's radius $R_E = 6378.145 \text{ km}$, the velocity by $\sqrt{R_E g_0}$, and the time by $\sqrt{R_E/g_0}$, where $g_0 = 9.807 \text{ m/s}^2$ is the gravi-

tational acceleration magnitude at the earth's radius R_E , and the mass by the vehicle's initial launch mass m_0 . According to Assumption 1, the vehicle's mass can be treated as an algebraic function of time instead of a state. Then, the dimensionless motion of ascent flight is given by

$$\begin{bmatrix} \mathbf{R}'_i \\ \mathbf{V}'_i \end{bmatrix} = \begin{bmatrix} \mathbf{V}_i \\ -\frac{\mu \mathbf{R}_i}{r_i^3} + \frac{\mathbf{D}_i}{m_i} + \frac{\mathbf{T}_i}{m_i} \mathbf{u}_i \end{bmatrix}, i = 1, 2 \quad (5)$$

$$m_i(t) = m_{i0} - q_i(t - t_{i0}), i = 1, 2 \quad (6)$$

where $(\cdot)' = d(\cdot)/dt$ with dimensional time t . To avoid confusion, the dimensionless variables in Equations (5) and (6) are denoted by the same symbols in Equation (1).

2.2. Ascent Trajectory Optimization Problem

The objective of the launch vehicle problem is to find the optimal thrust direction of the vehicle to accurately send the payload into the prescribed target orbit while minimizing fuel consumption. In this paper, the exhaustion cutoff strategy is utilized in the first flight phase of the TSLV, i.e., the flight time of phase 1 is determined by the total propellant of the first stage. Since the mass flow rate is assumed to be constant, and the rocket of phase 1 will shut down after the propellant is exhausted, the flight time t_{1f} is fixed. The flight time t_{2f} of phase 2 is free because the orbit insertion time cannot be determined in advance. We ignore the stage separation time duration and assume the start time of phase 2 $t_{20} = t_{1f}$.

The objective of minimizing fuel consumption is equivalent to minimizing flight time t_{2f} ; therefore, the cost function is to minimize.

$$J = t_{2f} \quad (7)$$

The boundary constraints at the initial time t_0 are determined by coordinates of the launch site, so that

$$\mathbf{R}_1(t_0) = \mathbf{R}_0, \quad \mathbf{V}_1(t_0) = \mathbf{V}_0 \quad (8)$$

The linkage between the two phases is constrained by

$$\mathbf{R}_1(t_{1f}) = \mathbf{R}_2(t_{20}), \quad \mathbf{V}_1(t_{1f}) = \mathbf{V}_2(t_{20}) \quad (9)$$

The terminal states are constrained by the orbital insertion conditions, which are determined by the prescribed orbital elements, i.e., the semi-major axis a , eccentricity e , inclination i , right ascension of ascending node Ω , and argument of perigee ω . In light of [30,31], the final position $\mathbf{R}_{2f}$ and velocity $\mathbf{V}_{2f}$ are constrained by the following conditions:

$$\begin{cases} a(t_{2f}) = \psi_a(\mathbf{R}_{2f}, \mathbf{V}_{2f}) = a_f \\ e(t_{2f}) = \psi_e(\mathbf{R}_{2f}, \mathbf{V}_{2f}) = e_f \\ i(t_{2f}) = \psi_i(\mathbf{R}_{2f}, \mathbf{V}_{2f}) = i_f \\ \Omega(t_{2f}) = \psi_\Omega(\mathbf{R}_{2f}, \mathbf{V}_{2f}) = \Omega_f \\ \omega(t_{2f}) = \psi_\omega(\mathbf{R}_{2f}, \mathbf{V}_{2f}) = \omega_f \end{cases} \triangleq \boldsymbol{\Psi}(\mathbf{R}_{2f}, \mathbf{V}_{2f}) = \boldsymbol{\Psi}_f \quad (10)$$

where the functions ψ_a , ψ_e , ψ_i , ψ_Ω , and ψ_ω are the relations between the inertial position and velocity and the respective orbital elements. The details of these functions can be found in Appendix C of [30].

To guarantee the thrust direction $\mathbf{u}_i$ is of the unit vector, a path constraint is imposed on the control input as

$$\|\mathbf{u}_i\| = 1. \quad (11)$$

Another path constraint is that the heat flux must not exceed a given value, such that the TSLV can safely undergo severe thermal conditions during the atmospheric flight. Hence, the following heat flux constraint is imposed only in phase 1 [2]:

$$\dot{Q} = \frac{1}{2} \rho v_{i \text{ rel}}^3 \leq \dot{Q}_{\max}, \quad i = 1, \quad (12)$$

The ascent trajectory optimization problem of the TSLV is formulated as the following two-phase optimal control problem P1 with free final time:

$$\begin{aligned} \text{P1 : } \min J &= t_{2f} \\ \text{s.t. } &(5), (6), (8), (9), (10), (11), (12) \end{aligned} \quad (13)$$

3. Convex Transcription

A convex programming problem is defined in a convex feasible convex set by a convex objective function, linear equality constraints, and inequality constraints. The ascent trajectory optimization problem P1 is nonlinear and non-convex. To use the convex programming method, the equivalent convex version of P1 is formulated in this section.

3.1. Constraints Relaxation

The path constraint Equation (11) is a nonlinear equality constraint that can be relaxed as the following inequalities:

$$\|u_i\| \leq 1 \quad (14)$$

which is a convex constraint, particularly a SOCP constraint. Then, we henceforth have the following relaxed problem P2:

$$\begin{aligned} \text{P2 : } \min J &= t_{2f} \\ \text{s.t. } &(5), (6), (8), (9), (10), (12), (14) \end{aligned} \quad (15)$$

Although the above convex relaxation renders a larger feasible set than the original one, the resulting optimal control problem P2 has the same solution as the original problem, and this can be guaranteed by the following proposition under a mild assumption:

Assumption 3. Constraint (12) is assumed to be inactive a.e. (almost everywhere) in $[t_{10}, t_{2f}]$.

Remark 2. Assumption 3 demonstrates that the heat flux constraint is typically active only at finite points in time during atmospheric flight. Actually, this assumption almost always holds for the ascent problem [2].

Proposition 1. If $\{R_i^*, V_i^*, u_i^*, t_{if}^*\} (i = 1, 2)$ is a solution of P2, then it is also a solution of P1 and $\|u_i^*\| = 1$ holds a.e. in $[t_{10}, t_{2f}]$.

Proof of Proposition 1. Proposition 1 is proved by contradiction following [32] by Liu et al. Rewrite the objective function of minimizing flight time as $J = \int_0^{t_f} 1 dt$, then according to the optimal control theory, the Hamiltonian H is given by

$$H = 1 + \lambda_R^T V + \lambda_V^T \left(-\frac{\mu R}{r^3} + \frac{D}{m} + \frac{T}{m} u \right) \quad (16)$$

where $\lambda = [\lambda_R^T \ \lambda_V^T \ \lambda_{t_f}]^T$ is the costate vector, and here we omit the subscript i representing the phase in Equation (5) for convenience, and such an omission does not affect the problem. Taking into account the path constraints, the Lagrangian L is

$$L = H - \lambda_Q(\dot{Q} - \dot{Q}_{\max}) - \lambda_u(\|u\| - 1) \quad (17)$$

where λ_Q and λ_u are the Lagrange multipliers corresponding to the constraints Equations (11) and (12), respectively. The optimal solution should satisfy the following complementary slack conditions:

$$\lambda_Q \geq 0, \quad \lambda_Q(\dot{Q} - \dot{Q}_{\max}) = 0 \quad (18)$$

$$\lambda_u \geq 0, \quad \lambda_u(\|u\| - 1) = 0 \quad (19)$$

The costate equations are given as

$$\dot{\lambda}_R = -\frac{\partial L}{\partial R} = \left(\frac{\mu}{r^3} I_3 - \frac{1}{m} \frac{\partial D}{\partial R} \right) \lambda_V + \lambda_Q \frac{\partial \dot{Q}}{\partial R} \quad (20)$$

$$\dot{\lambda}_V - \frac{\partial L}{\partial V} = -\lambda_R - \frac{1}{m} \frac{\partial D}{\partial V} \lambda_V + \lambda_Q \frac{\partial \dot{Q}}{\partial V} \quad (21)$$

According to Pontryagin's minimum principle, the optimal control u^* must be

$$u^* = \underset{u \in \mathcal{U}}{\operatorname{argmin}} H(R^*, V^*, u, \lambda, t) \quad (22)$$

where $\mathcal{U} = \{u : \|u\| \leq 1\}$. To minimize the Hamiltonian over $u \in \mathcal{U}$, the following condition should hold:

$$\frac{\partial L}{\partial u} = \frac{T}{m} \lambda_V - \lambda_u \frac{u}{\|u\|} = 0 \quad (23)$$

Because the terminal time t_f is free, the Lagrangian is null at the final time:

$$L(t_f) = 0 \quad (24)$$

Now, we start the proof by contradiction. Supposing there exists a finite interval where $\|u\| < 1$ holds, Equation (19) implies $\lambda_u = 0$; then, we henceforth have $\lambda_V = 0$. Meanwhile, the heat flux constraint is a.e. inactive based on Assumption 3, i.e., $\dot{Q} < \dot{Q}_{\max}$, then, $\lambda_Q = 0$ holds in Equation (18). From the costate equations in Equations (20) and (21), we have

$$\dot{\lambda}_R = 0, \quad \dot{\lambda}_V = -\lambda_R \quad (25)$$

Then, we henceforth have $\lambda_R = 0$, thus $L = 1$, and this violates Equation (24).

Based on the above analysis, we can conclude the $\|u_i^*\| = 1$ holds almost everywhere on the time interval $[t_{10}, t_{2f}]$. $\square$

3.2. Linearization

Since the dynamic constraints in Equation (5) are control-affine but still nonlinear [2], the linearization techniques should be used to obtain their corresponding convex constraints. To avoid the linearization computations on the aerodynamic terms during the sequential iterations, we can rewrite (5) as the following pseudo-linear form:

$$x_i' = f(x_i, u_i, t) = \tilde{A}_i(r_i)x_i + \tilde{B}u_i + \tilde{C}_i \quad (26)$$

where $\mathbf{x}_i = [\mathbf{R}_i^T, \mathbf{V}_i^T]^T$, $\tilde{\mathbf{A}}_i(r_i) = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{A}(r_i) & \mathbf{0}_{3 \times 3} \end{bmatrix}$ with $\mathbf{A}_i(r_i) = -\frac{\mu}{r_i^3} \mathbf{I}_{3 \times 3}$, $\tilde{\mathbf{B}}_i = \begin{bmatrix} \mathbf{0}_{3 \times 3} \\ \mathbf{B}_i \end{bmatrix}$ with $\mathbf{B}_i = \frac{T_i}{m_i} \mathbf{I}_{3 \times 3}$, and $\tilde{\mathbf{C}}_i = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{C}_i \end{bmatrix}$ with $\mathbf{C} = \mathbf{D}_i/m_i$. Here, $\mathbf{0}_{m \times n}$ and $\mathbf{I}_{m \times n}$ denote the zero and identity matrix of size $m \times n$, respectively. In Equation (26), we use the state-dependent coefficient matrix $\tilde{\mathbf{A}}_i(r_i)$ instead of $\tilde{\mathbf{A}}(\mathbf{x}, \mathbf{u}) = \partial \mathbf{f} / \partial \mathbf{x}$ in the classical linearization method for preserving the nonlinear characteristics. This makes the trust-region constraints unnecessary for guaranteeing that the linearized system sufficiently approximates the original nonlinear system [33]. The influence of the aerodynamic term $\tilde{\mathbf{C}}_i$ will be discussed in the latter.

In the framework of sequential convex optimization, we assume a reference solution $\{\mathbf{x}_i^{(k)}, \mathbf{u}_i^{(k)}, t_{if}^{(k)}\} (i = 1, 2)$ has been obtained by the k^{th} iteration; then, the parameter matrices $\tilde{\mathbf{A}}_i$, $\tilde{\mathbf{B}}_i$, and $\tilde{\mathbf{C}}_i$ can be calculated and treated as constant matrices, thus a pseudo-linear constraint.

Considering the free-time problem in phase 2 of TSLV and the necessity of the pseudospectral method utilized in the latter of this paper, we use an independent variable τ defined over a fixed domain $[-1, 1]$ to replace time t for each phase by using the mapping $t_i = (t_{if} - t_{i0})\tau/2 + (t_{if} + t_{i0})/2, (i = 1, 2)$. Thus, the time dilation s_i between t_i and τ is given by

$$s_i = \frac{dt_i}{d\tau} = \frac{t_{if} - t_{i0}}{2}, (i = 1, 2) \quad (27)$$

Then, the pseudo-linear Equation (26) can be expressed in terms of τ and approximated around a reference solution $\{\mathbf{x}_i^{(k)}, \mathbf{u}_i^{(k)}, t_{if}^{(k)}\} (i = 1, 2)$ as follows:

$$\mathbf{x}'_i := \frac{d\mathbf{x}_i}{d\tau} = s_i \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i, \tau) \approx s_i^{(k)} \left[\tilde{\mathbf{A}}_i^{(k)}(r_i^{(k)}) \mathbf{x}_i + \tilde{\mathbf{B}}_i^{(k)} \mathbf{u}_i + \tilde{\mathbf{C}}_i^{(k)} \right] + (s_i - s_i^{(k)}) \boldsymbol{\Sigma} \quad (28)$$

where $s_i^{(k)} = (t_{if}^{(k)} - t_{i0}^{(k)})/2$, and $\boldsymbol{\Sigma} = \tilde{\mathbf{A}}_i^{(k)}(r_i^{(k)}) \mathbf{x}_i^{(k)} + \tilde{\mathbf{B}}_i^{(k)} \mathbf{u}_i^{(k)}$. Note that $\tilde{\mathbf{C}}_i^{(k)}$ is treated as a constant term and can be calculated from the known previous iteration solution and the reasonability of this small trick will be detailed by Proposition 2 at the end of this section.

The linearization formula in Equation (28) differs from the method of [2] due to the utilization of the pseudo-linear system. In Equation (28), $\tilde{\mathbf{A}}_i^{(k)}(r_i^{(k)})$ and $\tilde{\mathbf{B}}_i^{(k)}$ do not depend on the reference solution control input $\mathbf{u}_i^{(k)}$, this provides enhanced robustness to the successive linearization sequences as intermediate controls can change significantly among the iterative solving [34].

In this paper, the final time of phase 1 is fixed so that $s_1 = s_1^{(k)}$; then, Equation (28) reduces to

$$\mathbf{x}'_1 \approx s_1 \left[\tilde{\mathbf{A}}_1^{(k)}(r_1^{(k)}) \mathbf{x}_1 + \tilde{\mathbf{B}}_1^{(k)} \mathbf{u}_1 + \tilde{\mathbf{C}}_1^{(k)} \right] \quad (29)$$

which is equivalent to Equation (26) in essence, and has no instability issues during the iterations. While the final time of phase 2 is free such that

$$\mathbf{x}'_2 \approx s_2^{(k)} \left[\tilde{\mathbf{A}}_2^{(k)}(r_2^{(k)}) \mathbf{x}_2 + \tilde{\mathbf{B}}_2^{(k)} \mathbf{u}_2 + \tilde{\mathbf{C}}_2^{(k)} \right] + (s_2 - s_2^{(k)}) \left[\tilde{\mathbf{A}}_2^{(k)}(r_2^{(k)}) \mathbf{x}_2^{(k)} + \tilde{\mathbf{B}}_2^{(k)} \mathbf{u}_2^{(k)} \right] \quad (30)$$

when $s_2 = s_2^{(k)}$, Equation (30) reduces to a similar form as Equation (29), but when s_2 diverges excessively from the reference value $s_2^{(k)}$, some extra constraints should be imposed to ensure convergence. In particular, a further trust region constraint is imposed on the duration of phase 2 to ensure the Taylor expansion around $s_2^{(k)}$ is effective; therefore,

$$\left| s_2 - s_2^{(k)} \right| = \frac{1}{2} \left| t_{2f} - t_{2f}^{(k)} \right| \leq \Delta t \quad (31)$$

In our experience, a suitable choice for Δ_t may guarantee iterative convergency, and the sequential convex optimization procedure will autonomously determine the optimal final time.

The terminal insertion conditions in Equation (10) are nonlinear and can be linearized as

$$\Psi(\mathbf{R}_{2f}^{(k)}, \mathbf{V}_{2f}^{(k)}) + \frac{\partial \Psi}{\partial \mathbf{R}_2}(\mathbf{R}_{2f}^{(k)}, \mathbf{V}_{2f}^{(k)}) (\mathbf{R}_{2f} - \mathbf{R}_{2f}^{(k)}) + \frac{\partial \Psi}{\partial \mathbf{V}_2}(\mathbf{R}_{2f}^{(k)}, \mathbf{V}_{2f}^{(k)}) \cdot (\mathbf{V}_{2f} - \mathbf{V}_{2f}^{(k)}) = \Psi_f \quad (32)$$

The analytical formulas of partial derivatives of the orbital elements Ψ with respect to the position and velocity vectors can be found in Appendix B of [35]. In this paper the Jacobian matrices $\partial \Psi / \partial \mathbf{R}_2$ and $\partial \Psi / \partial \mathbf{V}_2$ are obtained by numerical computation instead of complicated analytical derivations.

Another constraint to be linearized is the heat flux constraint in Equation (12), which can be approximated by

$$\dot{Q}(\mathbf{R}_1^{(k)}, \mathbf{V}_1^{(k)}) + \frac{\partial \dot{Q}}{\partial \mathbf{R}_1}(\mathbf{R}_1^{(k)}, \mathbf{V}_1^{(k)}) \cdot (\mathbf{R}_1 - \mathbf{R}_1^{(k)}) + \frac{\partial \dot{Q}}{\partial \mathbf{V}_1}(\mathbf{R}_1^{(k)}, \mathbf{V}_1^{(k)}) \cdot (\mathbf{V}_1 - \mathbf{V}_1^{(k)}) \leq \dot{Q}_{\max} \quad (33)$$

where the partial derivatives of the heat flux can be calculated by

$$\frac{\partial \dot{Q}}{\partial \mathbf{R}_1} = \frac{1}{2} \frac{d\rho}{d\mathbf{R}_1} v_{1rel}^3 + \frac{3}{2} \rho v_{1rel} \boldsymbol{\omega}_E \times \mathbf{V}_{1rel}, \quad \frac{d\rho}{d\mathbf{R}_1} = -\frac{1}{H_s} \rho_0 e^{-h/H_s} \frac{\mathbf{R}_1}{r_1} \quad (34)$$

$$\frac{\partial \dot{Q}}{\partial \mathbf{V}_1} = \frac{3}{2} \rho v_{1rel} \mathbf{V}_{1rel} \quad (35)$$

After the linearization, problem P2 can be transformed as the following sequential convex optimization problem P3:

$$\begin{aligned} \text{P3 : } \min J &= t_{2f} \\ \text{s.t. } & (29), (30), (6), (8), (9), (14), (31), (32), (33) \end{aligned} \quad (36)$$

In problem P3, we replaced the constraints in Equations (10) and (12) with their first-order Taylor expansion forms Equations (32) and (33), respectively. Such replacements have been proven to guarantee that the solution of P3 is also the solution of P2 by Liu, X. et al. in [36]. As for the replacements of the dynamic constraints in Equation (5) by Equations (29) and (30), we have the following proposition:

Proposition 2. *If the sequential convex iteration stopping criteria*

$$\max_{\tau} \left\| \mathbf{x}_i(\tau) - \mathbf{x}_i^{(k)}(\tau) \right\| \leq \delta, i = 1, 2 \quad (37)$$

where δ is a sufficiently small positive number, and is satisfied while solving problem P3, and the aerodynamics force terms in Equations (29) and (30) satisfy $\left\| \tilde{\mathbf{C}}_i^{(k)}(\tau) - \tilde{\mathbf{C}}_i(\tau) \right\| \leq \delta_{C_i}$, where δ_{C_i} is also a small positive number and depends on the choice of δ .

Proof of Proposition 2. Since Equation (37) is satisfied and δ is a sufficiently small number, $\max_{\tau} \left\| \mathbf{R}_i(\tau) - \mathbf{R}_i^{(k)}(\tau) \right\| \leq \delta$ and $\max_{\tau} \left\| \mathbf{V}_i(\tau) - \mathbf{V}_i^{(k)}(\tau) \right\| \leq \delta$ hold for $i = 1, 2$. Note that the aerodynamic force $\mathbf{D}_i$ continuously depends on the position $\mathbf{R}_i$ and the velocity $\mathbf{V}_i$ according to Equation (2), which is Lipschitz continuous. We henceforth have $\left\| \mathbf{D}_i(\tau) - \mathbf{D}_i^{(k)}(\tau) \right\| \leq \delta_D$ where δ_D is a small positive number that relates to δ , and this means $\left\| \tilde{\mathbf{C}}_i^{(k)}(\tau) - \tilde{\mathbf{C}}_i(\tau) \right\| \leq \delta_{C_i} = \delta_D$. $\square$

Remark 3. In ascent trajectory optimization, the strong coupling of the aerodynamic forces with the thrust direction and other path constraints makes the convergence of any algorithm difficult to achieve [4]. In this paper, we proposed a small trick using the drag force profile rendered from the solution of the last iteration as the current drag force profile, and linearizing the drag force around the reference trajectory is not required. The reasonability of this trick is theoretically demonstrated by Proposition 2, and the effectiveness will be validated in Section 6 by the convincing numerical results.

4. Discretization via Well-Conditioned Second-Order Birkhoff PS Method

As the final step, the convexified problem P3, which is still a continuous-time problem, must be transcribed into a convex programming problem with a finite set of decision variables and convex constraints to solve by numerical algorithms. Considering that the dynamics constraints in Equations (29) and (30) are second-order cascaded systems, the second-order pseudospectral method is used to transcribe the problem P3. In this section, a brief of the second-order Birkhoff interpolation is presented, and then, the well-conditioned convex programming problem is formulated by using the second-order Birkhoff pseudospectral method.

4.1. Brief of the Second-Order Birkhoff PS Method

In the conventional PS optimal method, any continuous-time signal $f(\tau)$, $\tau \in [-1, 1]$ can be approximated by

$$f(\tau) = \sum_{i=0}^N f_i(\tau_i) \mathcal{L}_i(\tau) \quad (38)$$

where $\tau_i, i = 0, 1, \dots, N$ are the discretizing collocation points, $-1 \leq \tau_0 < \tau_1 < \dots < \tau_{N-1} < \tau_N \leq 1$, and the orthogonal Lagrange interpolation polynomials $\mathcal{L}_i \in P_N$ is at the freedom of degree N satisfying the Kronecker relationship:

$$\mathcal{L}_j(\tau_i) = \delta_{ij} \quad (39)$$

There are several choices for $\mathcal{L}_i$, and the Chebyshev polynomials are used in this paper based on the works [27,28], in which, the Chebyshev–Gauss–Labotto (CGL) PS method performs with a better accuracy and efficiency than the Legendre–Gauss–Labotto (LGL) PS method in dealing with sequential convex optimization problems. When the discrete point set is the type of Gauss–Labotto (GL), the boundary points ± 1 are included such that $-1 = \tau_0 < \tau_1 < \dots < \tau_{N-1} < \tau_N = 1$.

Differentiating (26) with respect to time τ and evaluating $\dot{f}(\tau)$ on the discrete points yields

$$\dot{f}(\tau_i) = \sum_{j=0}^N f_j \dot{\mathcal{L}}_j(\tau_i) \triangleq \dot{\mathbf{f}} = \mathcal{D}\mathbf{f}, \quad i = 0, 1, \dots, N \quad (40)$$

where $\mathcal{D} \in \mathbb{R}^{(N+1) \times (N+1)}$, defined by the entry $d_{ij} = \dot{\mathcal{L}}_j(\tau_i)$ for $0 \leq i, j \leq N$, is the PS differentiation matrix (PSDM). By using Equation (28), one can transcribe the dynamic constraints in problem P3 into finite-dimensional constraints.

Differing from the above classical PS method, the second-order Birkhoff PS method is derived from the second-order Birkhoff interpolation polynomials as follows:

$$f(\tau) = f(\tau_0)B_0(\tau) + \sum_{i=1}^{N-1} \dot{f}(\tau_i)B_i(\tau) + f(\tau_N)B_N(\tau), \quad \tau \in [-1, 1] \quad (41)$$

where $B_i (i = 0, \dots, N)$ is the counterpart of the Lagrange basis polynomials $\{\mathcal{L}_i\}_{i=0}^N$. According to Theorem 3.1 in [37], we henceforth have

$$\begin{aligned} B_0(\tau) &= \frac{1-\tau}{2} \\ B_i(\tau) &= \frac{1+\tau}{2} \int_{-1}^1 (\tau-1) \tilde{\mathcal{L}}_i(\tau) d\tau + \int_{-1}^{\tau} (\tau-t) \tilde{\mathcal{L}}_i(t) dt, j = 1, \dots, N-1 \\ B_N(\tau) &= \frac{1+\tau}{2} \end{aligned} \quad (42)$$

where $\tilde{\mathcal{L}}_i (i = 1, \dots, N-1)$ are the Lagrange basis polynomials at the freedom of degree $N-2$.

Evaluating Equations (40) and (41) at the given GL points yields

$$\mathbf{f} = \mathcal{B} \tilde{\mathbf{f}}^{(2)} \quad (43)$$

where $\tilde{\mathbf{f}}^{(2)} = [f(\tau_0), \ddot{f}(\tau_1), \dots, \ddot{f}(\tau_{N-1}), f(\tau_N)]^T$, $\mathbf{f} = [f(\tau_0), f(\tau_1), \dots, f(\tau_{N-1}), f(\tau_N)]^T$, and the second-order Birkhoff PS integration matrix (BPSIM) $\mathcal{B}$ is given as

$$\begin{aligned} \mathcal{B} &= [B_j(\tau_i)]_{0 \leq i, j \leq N} \in \mathbb{R}^{(N+1) \times (N+1)} \\ \mathcal{B}_{in} &= [B_j(\tau_i)]_{1 \leq i, j \leq N-1} \in \mathbb{R}^{(N-1) \times (N-1)} \end{aligned} \quad (44)$$

Differentiating both sides of Equation (43) with respect τ and combining Equation (40) yields

$$\dot{\mathbf{f}} = \mathcal{D} \mathbf{f} = \mathcal{D} \mathcal{B} \tilde{\mathbf{f}}^{(2)} = \mathcal{B}^{(1)} \tilde{\mathbf{f}}^{(2)} \quad (45)$$

and differentiating again results in

$$\ddot{\mathbf{f}} = \mathcal{D} \mathcal{D} \mathbf{f} = \mathcal{D}^{(2)} \mathbf{f} = \mathcal{D}^{(2)} \mathcal{B} \tilde{\mathbf{f}}^{(2)} = \mathcal{B}^{(2)} \tilde{\mathbf{f}}^{(2)} \quad (46)$$

Note that the above relation between the BPSIM $\mathcal{B}$ and DPSM $\mathcal{D}$ provides an easy way to calculate $\mathcal{B}$. Replace the first row and the last row of $\mathcal{D}^{(2)}$ by $\mathbf{e}_1 = [1, 0, \dots, 0]$ and $\mathbf{e}_N = [0, \dots, 0, 1]$, respectively, therefore a matrix denoted by $\tilde{\mathcal{D}}^{(2)}$, so that $\tilde{\mathcal{D}}^{(2)} \mathbf{f} = \tilde{\mathbf{f}}^{(2)}$; further we have

$$\tilde{\mathcal{D}}^{(2)} \mathcal{B} = \mathbf{I}_{(N+1) \times (N+1)}, \quad \mathcal{D}_{in}^{(2)} \mathcal{B}_{in} = \mathbf{I}_{(N-1) \times (N-1)} \quad (47)$$

Thus, $\mathcal{B}$ can be obtained by the inverse of $\tilde{\mathcal{D}}^{(2)}$.

4.2. Second-Order Birkhoff PS Transcription

To use the second-order Birkhoff PS method to transcribe the problem P3, we rewrite Equations (29) and (30) as the following second-order form:

$$\mathbf{R}_1'' = \frac{d^2}{d\tau^2} \mathbf{R}_1 \approx s_1^2 \left[\mathbf{A}_1^{(k)}(r_1^{(k)}) \mathbf{R}_1 + \mathbf{B}_1^{(k)} \mathbf{u}_1 + \mathbf{C}_1^{(k)} \right] \quad (48)$$

$$\mathbf{R}_2'' = \frac{d^2}{d\tau^2} \mathbf{R}_2 \approx \left(s_2^{(k)} \right)^2 \left[\mathbf{A}_2^{(k)}(r_2^{(k)}) \mathbf{R}_2 + \mathbf{B}_2^{(k)} \mathbf{u}_2 + \mathbf{C}_2^{(k)} \right] + (s_2 - s_2^{(k)}) \left[\mathbf{A}_2^{(k)}(r_2^{(k)}) \mathbf{R}_2^{(k)} + \mathbf{B}_2^{(k)} \mathbf{u}_2^{(k)} \right] \quad (49)$$

Denote the unknown optimization decision variables over the discrete point set as

$$\mathcal{V}_i = \begin{bmatrix} \mathbf{R}_{i0}^T \\ \ddot{\mathbf{R}}_{i1}^T \\ \vdots \\ \ddot{\mathbf{R}}_{i(N_i-1)}^T \\ \mathbf{R}_{iN_i}^T \end{bmatrix} = \begin{bmatrix} \mathbf{R}_i^T(\tau_0) \\ \ddot{\mathbf{R}}_i^T(\tau_1) \\ \vdots \\ \ddot{\mathbf{R}}_i^T(\tau_{N_i-1}) \\ \mathbf{R}_i^T(\tau_{N_i}) \end{bmatrix} \in \mathbb{R}^{(N_i+1) \times 3}, i = 1, 2 \quad (50)$$

where $N_i + 1$ is the number of collocation points of the phase i . Let

$$\mathcal{V}_i^{in} = \begin{bmatrix} \ddot{\mathbf{R}}_{i1}^T \\ \vdots \\ \ddot{\mathbf{R}}_{i(N_i-1)}^T \end{bmatrix} \in \mathbb{R}^{(N_i-1) \times 3} \quad (51)$$

and $\mathcal{V}_{i0} = \mathbf{R}_{i0}^T, \mathcal{V}_{iN_i} = \mathbf{R}_{iN_i}^T$. According to Equations (42), (44), and (45), we have

$$\mathbf{R}_i = \begin{bmatrix} \mathbf{R}_{i0}^T \\ \mathbf{R}_{i1}^T \\ \vdots \\ \mathbf{R}_{i(N_i-1)}^T \\ \mathbf{R}_{iN_i}^T \end{bmatrix} = \begin{bmatrix} \mathbf{R}_i^T(\tau_0) \\ \mathbf{R}_i^T(\tau_1) \\ \vdots \\ \mathbf{R}_i^T(\tau_{N_i-1}) \\ \mathbf{R}_i^T(\tau_{N_i}) \end{bmatrix} = \mathcal{B}_i \mathcal{V}_i = \mathcal{B}_i \begin{bmatrix} \mathcal{V}_{i0} \\ \mathcal{V}_i^{in} \\ \mathcal{V}_{iN_i} \end{bmatrix} \quad (52)$$

$$\mathcal{B}_i^{(1)} \mathcal{V}_i = \mathcal{D}_i \mathcal{B}_i \mathcal{V}_i = \mathcal{D}_i \mathbf{R}_i = \dot{\mathbf{R}}_i = s_i \mathbf{V}_i \quad (53)$$

$$\mathcal{B}_i^{(2)} \mathcal{V}_i = \mathcal{D}_i^2 \mathcal{B}_i \mathcal{V}_i = \mathcal{D}_i^2 \mathbf{R}_i = \ddot{\mathbf{R}}_i \quad (54)$$

where the PSDM $\mathcal{D}_i$ and the BPSIM $\mathcal{B}_i$ are decided by the prescribed N_i for the discretization of phase i . From Equations (51) and (52), the vehicle's position and velocity vectors can be algebraically represented by the decision variables $\mathcal{V}_i$. Thus, the constraints in Equations (8) and (9) can be transformed as

$$\mathcal{V}_{10} = \mathbf{R}_{10}^T, \mathcal{B}_{10}^{(1)} \mathcal{V}_1 / s_1 = \mathbf{V}_{10}^T \quad (55)$$

$$\mathcal{V}_{1N_1} = \mathcal{V}_{20}, \mathcal{B}_{1N_1}^{(1)} \mathcal{V}_1 / s_1 = \mathcal{B}_{20}^{(1)} \mathcal{V}_2 / s_2^{(k)} \quad (56)$$

The final position and velocity vectors are given as

$$\mathcal{V}_{2N_2} = \mathbf{R}_{2f}^T, \mathcal{B}_{2N_2}^{(1)} \mathcal{V}_2 / s_2^{(k)} = \mathbf{V}_{2f}^T \quad (57)$$

Note that $\mathcal{B}_{i0}^{(1)}$ and $\mathcal{B}_{iN_i}^{(1)}$ stand for the first row and the last row of $\mathcal{B}_i^{(1)}$.

Let $\mathbf{U}_i = [\mathbf{u}_{i0}, \mathbf{u}_{i1}, \dots, \mathbf{u}_{iN_i}] = [\mathbf{u}_i(\tau_0), \mathbf{u}_i(\tau_1), \dots, \mathbf{u}_i(\tau_{N_i})] \in \mathbb{R}^{3 \times N_i} (i = 1, 2)$ be the control variables to be optimized. Then, the dynamic constraints in Equation (48) can be transcribed by

$$\begin{aligned} \mathcal{B}_{10}^{(2)} \mathcal{V}_1 &= \mathcal{F}_{10}(\mathbf{R}_{10}, \dot{\mathbf{R}}_{10}, \mathbf{u}_{10}) = s_1^2 \left[\mathbf{A}_{10}^{(k)} \mathbf{R}_{10} + \mathbf{B}_{10}^{(k)} \mathbf{u}_{10} + \mathbf{C}_{10}^{(k)} \right]^T \\ \mathcal{V}_1^{in} &= \mathcal{F}_{1in}(\mathbf{R}_{1in}, \dot{\mathbf{R}}_{1in}, \mathbf{u}_{1in}) = s_1^2 \begin{bmatrix} \left[\mathbf{A}_{11}^{(k)} \mathbf{R}_{11} + \mathbf{B}_{11}^{(k)} \mathbf{u}_{11} + \mathbf{C}_{11}^{(k)} \right]^T \\ \vdots \\ \left[\mathbf{A}_{1(N_i-1)}^{(k)} \mathbf{R}_{1(N_i-1)} + \mathbf{B}_{1(N_i-1)}^{(k)} \mathbf{u}_{1(N_i-1)} + \mathbf{C}_{1(N_i-1)}^{(k)} \right]^T \end{bmatrix} \\ \mathcal{B}_{1N_i}^{(2)} \mathcal{V}_1 &= \mathcal{F}_{1N_i}(\mathbf{R}_{1N_i}, \dot{\mathbf{R}}_{1N_i}, \mathbf{u}_{1N_i}) = s_1^2 \left[\mathbf{A}_{1N_i}^{(k)} \mathbf{R}_{1N_i} + \mathbf{B}_{1N_i}^{(k)} \mathbf{u}_{1N_i} + \mathbf{C}_{1N_i}^{(k)} \right]^T \end{aligned} \quad (58)$$

where $\mathcal{B}_{10}$ and $\mathcal{B}_{1N_1}$, respectively, are the first and the last row of $\mathcal{B}_1$, and $\mathbf{A}_{1j}^{(k)} = \mathbf{A}_1(r_j^{(k)})$ for $(j = 0, \dots, N_1)$ can be obtained by using the solution of the k th iteration. Similarly, the dynamics of phase 2 in Equation (49) is transcribed by

$$\begin{aligned} \mathcal{B}_{20}^{(2)} \mathcal{V}_2 &= \mathcal{F}_{20}(\mathbf{R}_{20}, \dot{\mathbf{R}}_{20}, \mathbf{U}_{20}) = (s_2^{(k)})^2 \left[\mathbf{A}_{20}^{(k)} \mathbf{R}_{20} + \mathbf{B}_{20}^{(k)} \mathbf{u}_{20} + \mathbf{C}_{20}^{(k)} \right]^T + (s_2 - s_2^{(k)}) \left[\mathbf{A}_{20}^{(k)} \mathbf{R}_{20}^{(k)} + \mathbf{B}_{20}^{(k)} \mathbf{u}_{20}^{(k)} \right]^T \\ \mathcal{B}_{20}^{(2)} \mathcal{V}_2 &= \mathcal{F}_{2in}(\mathbf{R}_{2in}, \dot{\mathbf{R}}_{2in}, \mathbf{U}_{2in}) \\ &= \begin{bmatrix} (s_2^{(k)})^2 \left[\mathbf{A}_{21}^{(k)} \mathbf{R}_{21} + \mathbf{B}_{21}^{(k)} \mathbf{u}_{21} + \mathbf{C}_{21}^{(k)} \right]^T + (s_2 - s_2^{(k)}) \left[\mathbf{A}_{21}^{(k)} \mathbf{R}_{21}^{(k)} + \mathbf{B}_{21}^{(k)} \mathbf{u}_{21}^{(k)} \right]^T \\ \vdots \\ (s_2^{(k)})^2 \left[\mathbf{A}_{2(N_2-1)}^{(k)} \mathbf{R}_{2(N_2-1)} + \mathbf{B}_{2(N_2-1)}^{(k)} \mathbf{u}_{2(N_2-1)} + \mathbf{C}_{2(N_2-1)}^{(k)} \right]^T + (s_2 - s_2^{(k)}) \left[\mathbf{A}_{2(N_2-1)}^{(k)} \mathbf{R}_{2(N_2-1)}^{(k)} + \mathbf{B}_{2(N_2-1)}^{(k)} \mathbf{u}_{2(N_2-1)}^{(k)} \right]^T \end{bmatrix}^T \\ \mathcal{B}_{2N_2}^{(2)} \mathcal{V}_2 &= \mathcal{F}_{2N_i}(\mathbf{R}_{2N_2}, \dot{\mathbf{R}}_{2N_2}, \mathbf{U}_{2N_2}) = (s_2^{(k)})^2 \left[\mathbf{A}_{2N_2}^{(k)} \mathbf{R}_{2N_2} + \mathbf{B}_{2N_2}^{(k)} \mathbf{u}_{2N_2} + \mathbf{C}_{2N_2}^{(k)} \right]^T + (s_2 - s_2^{(k)}) \left[\mathbf{A}_{2N_2}^{(k)} \mathbf{R}_{2N_2}^{(k)} + \mathbf{B}_{2N_2}^{(k)} \mathbf{u}_{2N_2}^{(k)} \right]^T \end{aligned} \quad (59)$$

The linearized orbital insertion conditions in Equation (32) can be reformulated as

$$\begin{aligned} \Psi(\mathbf{R}_{2f}^{(k)}, \mathbf{V}_{2f}^{(k)}) + \frac{\partial \Psi}{\partial \mathbf{R}_2}(\mathbf{R}_{2f}^{(k)}, \mathbf{V}_{2f}^{(k)}) \left((\mathcal{B}_{2N_2} \mathcal{V}_2)^T - \mathbf{R}_{2f}^{(k)} \right) \\ + \frac{\partial \Psi}{\partial \mathbf{V}_2}(\mathbf{R}_{2f}^{(k)}, \mathbf{V}_{2f}^{(k)}) \cdot \left((\mathcal{B}_{2N_2}^{(1)} \mathcal{V}_2)^T - \mathbf{V}_{2f}^{(k)} \right) = \Psi_f \end{aligned} \quad (60)$$

Similarly, the linearized heat flux constraint in Equation (33) can be rewritten as

$$\begin{aligned} \dot{Q}(\mathbf{R}_{1j}^{(k)}, \mathbf{V}_{1j}^{(k)}) + \frac{\partial \dot{Q}}{\partial \mathbf{R}_1}(\mathbf{R}_{1j}^{(k)}, \mathbf{V}_{1j}^{(k)}) \cdot \left((\mathcal{B}_{1j} \mathcal{V}_1)^T - \mathbf{R}_{1j}^{(k)} \right) \\ + \frac{\partial \dot{Q}}{\partial \mathbf{V}_1}(\mathbf{R}_{1j}^{(k)}, \mathbf{V}_{1j}^{(k)}) \cdot \left((\mathcal{B}_{1j}^{(1)} \mathcal{V}_1)^T / s_1 - \mathbf{V}_{1j}^{(k)} \right) \leq \dot{Q}_{\max}, j = 0, \dots, N_1 \end{aligned} \quad (61)$$

Collecting the above relevant equations, the convexified problem P3 is discretized as the following problem BPSC2:

$$\begin{aligned} \text{Decision variables : } \mathcal{V}_i &\in \mathbb{R}^{(N_i+1) \times 3}, \mathbf{U}_i \in \mathbb{R}^{3 \times N_i} \\ \text{Temporal variables : } \mathbf{R}_i &= \mathcal{B}_i \mathcal{V}_i, \mathbf{V}_i = \mathcal{B}_i^{(1)} \mathcal{V}_i \\ \text{BPSC2 : } \min J &= -t_{2f} \\ \text{s.t. } &(58), (59), (6), (55), (56), (14), (31), (60), (61) \end{aligned} \quad (62)$$

where the temporal variables can be algebraically represented by the decision variables $\mathcal{V}_i$, and facilitate us in establishing problem P4.

In problem BPSC2, the decision variables are not directly the states of the dynamic system at the collocation points as in the conventional PS method, but the variables relating to the half of states. Thus, the number of decision variables is dramatically reduced.

In the works [27,28], the discretization techniques DPSC1 by using the differential matrix $\mathcal{D}$ and BPCS1 by using the first-order Birkhoff PS method in the framework of sequential convex optimization have been well studied, and the readers can find the details in these works. To conduct comparison studies in Section 6, we briefly present the DPSC1 here. Let

$$\mathbf{X}_i = \begin{bmatrix} \mathbf{R}_{10}^T & \mathbf{V}_{10}^T \\ \vdots & \vdots \\ \mathbf{R}_{1N_i}^T & \mathbf{V}_{1N_i}^T \end{bmatrix} \in \mathbb{R}^{N_i \times 6}$$

and $\mathbf{U}_i \in \mathbb{R}^{3 \times N_i} (i = 1, 2)$ be the decision variables to be optimized, then we can directly use Equation (40) to transcribe Equations (29) and (30) as

$$\mathcal{D}_1 \mathbf{X}_1 = s_1 \begin{bmatrix} \left[\tilde{\mathbf{A}}_{10}^{(k)} \mathbf{X}_{10}^T + \tilde{\mathbf{B}}_{10}^{(k)} \mathbf{u}_{10} + \tilde{\mathbf{C}}_{10}^{(k)} \right]^T \\ \vdots \\ \left[\tilde{\mathbf{A}}_{1N_1}^{(k)} \mathbf{X}_{1N_1}^T + \tilde{\mathbf{B}}_{1N_1}^{(k)} \mathbf{u}_{1N_1} + \tilde{\mathbf{C}}_{1N_1}^{(k)} \right]^T \end{bmatrix} \quad (63)$$

$$\mathcal{D}_2 \mathbf{X}_2 = s_2^{(k)} \begin{bmatrix} \left[\tilde{\mathbf{A}}_{20}^{(k)} \mathbf{X}_{20}^T + \tilde{\mathbf{B}}_{20}^{(k)} \mathbf{u}_{20} + \tilde{\mathbf{C}}_{20}^{(k)} \right]^T \\ \vdots \\ \left[\tilde{\mathbf{A}}_{2N_2}^{(k)} \mathbf{X}_{2N_2}^T + \tilde{\mathbf{B}}_{2N_2}^{(k)} \mathbf{u}_{2N_2} + \tilde{\mathbf{C}}_{2N_2}^{(k)} \right]^T \end{bmatrix} + (s_2 - s_2^{(k)}) \begin{bmatrix} \left[\tilde{\mathbf{A}}_{20}^{(k)} \mathbf{x}_{20}^{(k)} + \tilde{\mathbf{B}}_{20}^{(k)} \mathbf{u}_{20}^{(k)} \right]^T \\ \vdots \\ \left[\tilde{\mathbf{A}}_{2N_2}^{(k)} \mathbf{x}_{2N_2}^{(k)} + \tilde{\mathbf{B}}_{2N_2}^{(k)} \mathbf{u}_{2N_2}^{(k)} \right]^T \end{bmatrix} \quad (64)$$

and the path constraints in Equation (61) can be rewritten as

$$\begin{aligned} \dot{\mathbf{Q}}(\mathbf{R}_{1j}^{(k)}, \mathbf{V}_{1j}^{(k)}) + \frac{\partial \dot{\mathbf{Q}}}{\partial \mathbf{R}_1}(\mathbf{R}_{1j}^{(k)}, \mathbf{V}_{1j}^{(k)}) \cdot (\mathbf{R}_{1j} - \mathbf{R}_{1j}^{(k)}) \\ + \frac{\partial \dot{\mathbf{Q}}}{\partial \mathbf{V}_1}(\mathbf{R}_{1j}^{(k)}, \mathbf{V}_{1j}^{(k)}) \cdot (\mathbf{V}_{1j} - \mathbf{V}_{1j}^{(k)}) \leq \dot{\mathbf{Q}}_{\max}, j = 0, \dots, N_1 \end{aligned} \quad (65)$$

Then, the DPSC1 is formulated as follows:

$$\begin{aligned} \text{Decision variables : } \mathbf{X}_i \in \mathbb{R}^{(N_i+1) \times 6}, \mathbf{U}_i \in \mathbb{R}^{3 \times N_i} \\ \text{DPSC1 : } \min J = -t_{2f} \\ \text{s.t. (63), (64), (6), (55), (56), (14), (31), (32), (65)} \end{aligned} \quad (66)$$

where the constraints in Equations (55)–(57) only need a slight change to accommodate the decision variables $\mathbf{X}_i$.

4.3. Sparsity Analysis of Convex Programming Problem

Problem BPSC2 as well as DPSC1 can be formulated as the following standard SOCP problem:

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & \mathcal{A} \mathbf{x} = \mathbf{b} \\ & \mathcal{G} \mathbf{x} + \mathbf{s} = \mathbf{h}, \mathbf{s} \in \mathcal{K} \end{aligned} \quad (67)$$

where $\mathbf{x}$ are decision variables to be optimized, $\mathbf{s}$ are the slack variables, $\mathcal{A}$, $\mathcal{G}$, $\mathbf{c}$, $\mathbf{b}$, and $\mathbf{h}$ depend on the problem, and $\mathcal{K}$ are conic constraints. The matrices $\mathcal{A}$ and $\mathcal{G}$ are often sparse, and their numerical condition and sparsity patterns will dramatically affect the efficiency of any solver algorithms. In [27], the condition number of $\mathcal{A}$ rendered by BPSC2 is much smaller than that by DPSC1, thereby giving a better performance on computational cost.

In this paper, since the heat flux constraints are introduced in BPSC2, the sparsity pattern of $\mathcal{G}$ rendered by BPSC2 becomes more complex as shown in Figure 1a,b, in which the discretization collocation point numbers are set as $N_1 = N_2 = 30$. Figure 2a,b present the sparsity patterns of $\mathcal{A}$ rendered by DPSC1 and BPSC2, respectively. The number of nonzero elements of $\mathcal{A}$ rendered by BPSC2 only is nearly half of that by DPSC1. Unfortunately, the BPSC2 method leads to a great increase in the number of nonzero elements of $\mathcal{G}$. This seriously deteriorates the sparsity pattern of $\mathcal{G}$. In fact, the more nonzero elements, the more storage and computation cost. According to our numerical experiments in Section 6, the sparsity patterns of $\mathcal{G}$ impose a great effect on the algorithm's efficiency, and the more dense $\mathcal{G}$ is, the longer the solving time, although the number of nonzero elements of $\mathcal{A}$ reduced by nearly half.

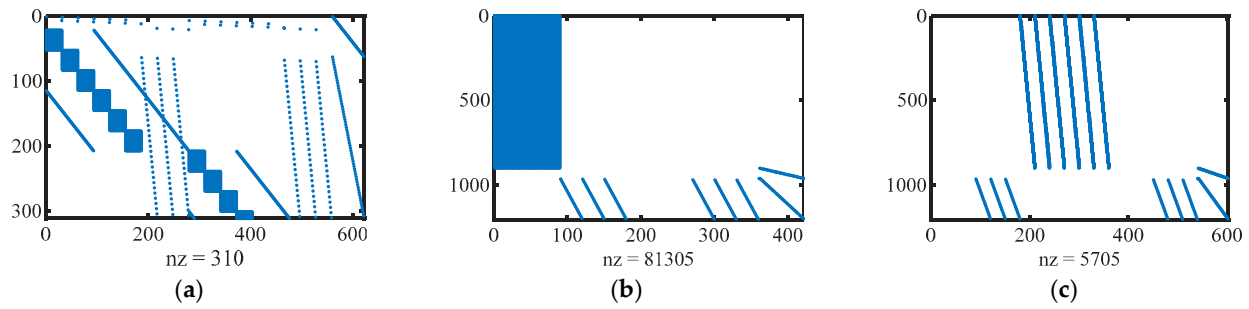


Figure 1. Sparsity patterns of $\mathcal{G}$ when $N_1 = N_2 = 30$: (a) sparsity pattern of $\mathcal{G}$ rendered by DBSC1; (b) by BPSC2; and (c) by A-BPSC2.

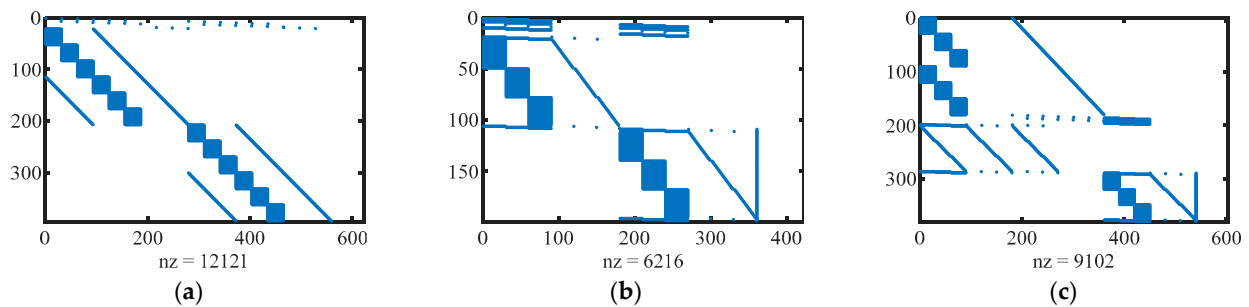


Figure 2. Sparsity patterns of $\mathcal{A}$ when $N_1 = N_2 = 30$: (a) sparsity pattern of $\mathcal{A}$ rendered by DPSC1; (b) by BPSC2; and (c) by A-BPSC2.

Comparing Equations (61) and (65), it can be found that the temporal variables $\mathbf{R}_i = \mathcal{B}_i \mathcal{V}_i$ and $\mathbf{V}_i = \mathcal{B}_i^{(1)} \mathcal{V}_i$ make that the dense matrices $\mathcal{B}_i$ and $\mathcal{B}_i^{(1)}$ are the necessary parts of $\mathcal{G}$; to overcome this shortage of BPSC2, we introduce new decision variables $\mathbf{R}_i, \mathbf{V}_i \in \mathbb{R}^{(N_i+1) \times 3}$ to formulate the path constraints in Equation (65); meanwhile, the algebraical relations $\mathbf{R}_i = \mathcal{B}_i \mathcal{V}_i$ and $\mathbf{V}_i = \mathcal{B}_i^{(1)} \mathcal{V}_i$ become additional equality constraints, thereby giving an augmented BPSC2, namely A-BPSC2, as follows:

$$\begin{aligned}
 &\text{Decision variables : } \mathcal{V}_i \in \mathbb{R}^{(N_i+1) \times 3}, \mathbf{U}_i \in \mathbb{R}^{3 \times N_i}; \mathbf{R}_i, \mathbf{V}_i \in \mathbb{R}^{(N_i+1) \times 3} \\
 &\text{A - BPSC2 : } \min J = -t_{2f} \\
 &\quad \text{s.t. } \mathbf{R}_i = \mathcal{B}_i \mathcal{V}_i, \mathbf{V}_i = \mathcal{B}_i^{(1)} \mathcal{V}_i \\
 &\quad (58), (59), (6), (55), (56), (14), (31), (32), (65)
 \end{aligned} \tag{68}$$

Remark 4. In problem A-BPSC2, the augmented decision variables $\mathbf{R}_i$ and $\mathbf{V}_i$ make the nonzero elements of $\mathcal{A}$ slightly increase, but the nonzero elements of $\mathcal{G}$ dramatically decrease (shown in Figures 1c and 2c). Thus, A-BPSC2 makes the matrix $\mathcal{G}$ sparser at a small cost, compromising the size of $\mathcal{A}$ and the sparsity pattern of $\mathcal{G}$. The advantages of A-BPSC2 over BPSC2 will be validated by the latter numerical results.

5. Solution Strategy

After convexification and discretization, the original non-convex trajectory optimization problem of TSLV becomes a series of SOCP problems, which can be iteratively solved by the efficient convex optimization toolbox, such as ECOS and Mosek. In fact, the iterative solving procedure should start from an initial reference trajectory, and its choice apparently affects the convergence of the algorithm. However, it is hard to guess an appropriate three-dimensional reference trajectory including the profile of three-dimensional position

and velocity, especially for the complex flight trajectory of TSLVs. To overcome this issue, we propose the solving procedure, which consists of two sub-procedures:

1. In the first iteration, taking no account of the aerodynamic drag, the simplified version of A-BPSC2 (or BPSC2, or DPCS1) with fixed final time is formulated based on the two guesses about the insertion time t_{2f} and the radius of the separation point r_{1f} of the vehicle as well as the known initial radius r_{10} and the final radius r_{2f} ; then, by using a convex solver, we have a vacuum solution;
2. Based on the vacuum solution provided by the above sub-procedure, the complete problem of A-BPSC2 (or BPSC2, or DPCS1) can be formulated, and then, the formal sequential convex iterative solving procedure begins until the iteration stopping criteria (26) are satisfied.

Using P stands for the three methods in this paper, i.e., A-BPSC2, BPSC2, and DPCS1, the above procedure is summarized in Algorithm 1.

Algorithm 1. SC Algorithm for ascent flight trajectory optimization

Algorithm parameters setting:

- The number of collocation points $N_i (i = 1, 2)$;
 - The iteration-stopping criteria δ ;
 - The trust region Δ_i ;
 - The number of max iterations $K_{\max}$
-

Problem initialization:

- The initial state $\mathbf{x}_{10} = [\mathbf{R}_0^T, \mathbf{V}_0^T]^T$;
 - The terminal state $\mathbf{x}_{2f} = [\mathbf{R}_{2f}^T, \mathbf{V}_{2f}^T]^T$;
 - The separation time $t_{1f} = t_{20}$;
 - The guess of the radius of the separation point $r_{1f}^{(0)} = r_{20}^{(0)}$;
 - The guess of the orbital insertion time $t_{2f}^{(0)}$
-

Sub-procedure 1:

1. Calculate the initial radius profile $r_{ij}^{(0)} (i = 1, 2, j = 1, \dots, N_i)$ via linear interpolation on $[r_{10}^{(0)}, r_{1f}^{(0)}]$ and $[r_{20}^{(0)}, r_{2f}^{(0)}]$;
 2. Formulate the problem $P^{(0)}$ based on $r_{ij}^{(0)} (i = 1, 2, j = 1, \dots, N_i)$ disregarding the aerodynamic drag D ;
 3. Solving the problem $P^{(0)}$ yields the reference trajectory $\{\mathbf{x}_i^{(1)}, \mathbf{u}_i^{(1)}, t_{if}^{(1)}\} (i = 1, 2)$
-

Sub-procedure 2:

Set $k = 1$;

While $k \leq K_{\max}$

Formulate the complete problem $P^{(k)}$ along $\{\mathbf{x}_i^{(k)}, \mathbf{u}_i^{(k)}, t_i^{(k)}\} (i = 1, 2)$;

Solving the problem $P^{(k)}$ yields $\{\mathbf{x}_i^{(k+1)}, \mathbf{u}_i^{(k+1)}, t_{if}^{(k+1)}\} (i = 1, 2)$;

If $\max_{\tau_j} \|\mathbf{x}_i^{(k+1)}(\tau_j) - \mathbf{x}_i^{(k)}(\tau_j)\| \leq \delta (i = 1, 2, j = 0, \dots, N_i)$

Output the optimal solution $\{\mathbf{x}_i^*, \mathbf{u}_i^*, t_i^*\} = \{\mathbf{x}_i^{(k+1)}, \mathbf{u}_i^{(k+1)}, t_{if}^{(k+1)}\} (i = 1, 2)$;

Break;

Else

Let $k = k + 1$, and $\{\mathbf{x}_i^{(k)}, \mathbf{u}_i^{(k)}, t_i^{(k)}\} = \{\mathbf{x}_i^{(k+1)}, \mathbf{u}_i^{(k+1)}, t_{if}^{(k+1)}\} (i = 1, 2)$

Continue;

End If

End While

6. Numerical Results

In this section, the numerical results are presented to demonstrate the effectiveness of the proposed algorithm by using the second-order Birkhoff PS scheme as well as the sequential convex programming method, denoted by BPSC2. YALMIP [38], a MATLAB R2021b toolbox, is used for modeling the sequential convex problems, and the light-embedded toolkit ECOS [39,40] based on the popular primal-dual interior-point algorithm is employed to solve the SOCP programming problem. The following numerical results including the graphics are obtained by MATLAB R2021b running on a desktop computer with Intel Xeon Gold 5218R CPU@2.10 GHz and 128 GB RAM. It is to be noted that the required memory for solving such trajectory optimization problems is much smaller than 128 GB.

According to the public data of China's Long March 7, the parameters used to model the TSLV are given in Table 1. The latitude and the longitude of the launch site, respectively, are $N19.61^\circ$ and $E110.95^\circ$; therefore, the initial position is $\mathbf{R}_{10}$ and the velocity is $\mathbf{V}_{10}$. The target orbit is a low earth orbit (LEO) of $200 \text{ km} \times 400 \text{ km}$, and the perigee is chosen as the insertion point for the precision control of the orbital plane and altitude. The prescribed inclination, right ascension of ascending node, and argument of perigee, respectively, are 42° , 312.9° , and 168.3° , the desired orbital semi-major axis is $6.6781399 \times 10^6 \text{ m}$, and eccentricity of the target orbit is 0.015.

Table 1. TSLV data.

Quantity	Stage 1	Stage 2	Unit
Initial mass m_{i0}	575,637	111,181	kg
Thrust T_i	7195	720	kN
Mass flow rate q_i	2447	215	kg/s
Reference area S_{ref}	12.5	12.5	m^2
Drag coefficient C_D	0.5	0.5	/
Max heat flux $\dot{Q}_{\max}$	4.0×10^7	/	W/m^2



The algorithm parameters are set as the iteration-stopping criteria $\delta = 10^{-5}$, the trust region $\Delta_t = 5\text{s}$, and the max iteration $K_{\max} = 30$. The flight time of phase 1 is $t_{1f} = t_{20} = 166\text{s}$, which is fixed, and the guess of the insertion time is $t_{2f}^{(0)} = 590\text{s}$. The height of the separation point is guessed as 56 km, i.e., $r_{1f}^{(0)} = r_{20}^{(0)} = R_E + 56(\text{km})$.

To verify the proposed method denoted by BPSC2, the famous NLP trajectory optimization toolbox GPOPS [41] is used as a baseline method; meanwhile, comparison studies between the proposed A-BPSC2, BPSC2, and DPSC1 are conducted. Figure 3 shows the height and velocity profile of the optimal ascent trajectory of TSLV resulting from GPOPS, DPSC1, BPSC2, and A-BPSC2, and Figure 4 illustrates the 3D position profile and the 3D velocity profile in the ECI frame. The optimal thrust directions are presented in Figure 5, and it can be observed that $\|\mathbf{u}\| = 1$ holds almost everywhere in the time interval $[t_{10}, t_{2f}]$ while the relaxed constraint $\|\mathbf{u}\| \leq 1$ is used in A-BPSC2, BPSC2, and DPSC1. Although the strict equality constraint $\|\mathbf{u}\| = 1$ is used in GPOPS, the resulting control inputs do not strictly satisfy the constraints. On the contrary, the results of the three SC algorithms reveal a better performance, and this well-verified Proposition 1.

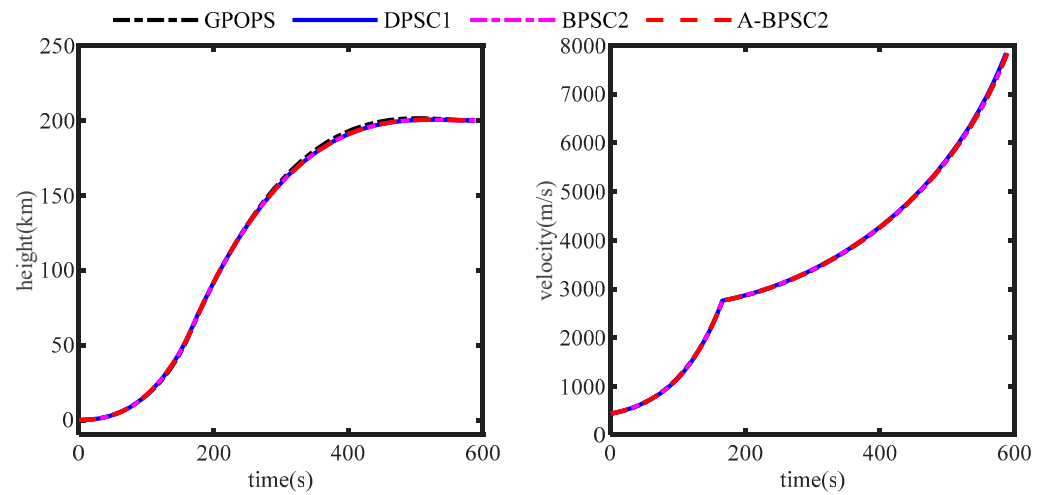


Figure 3. Optimized ascent trajectory of TSLV when $N_1 = N_2 = 31$: (Left): height profile; (right): velocity profile.

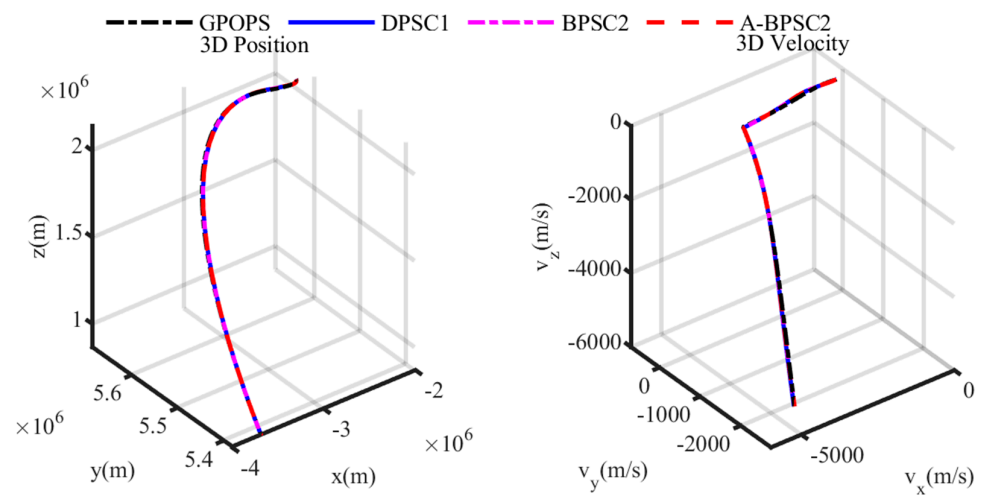


Figure 4. 3D trajectory of TSLV: (left): 3D position profile; (right): 3D velocity profile.

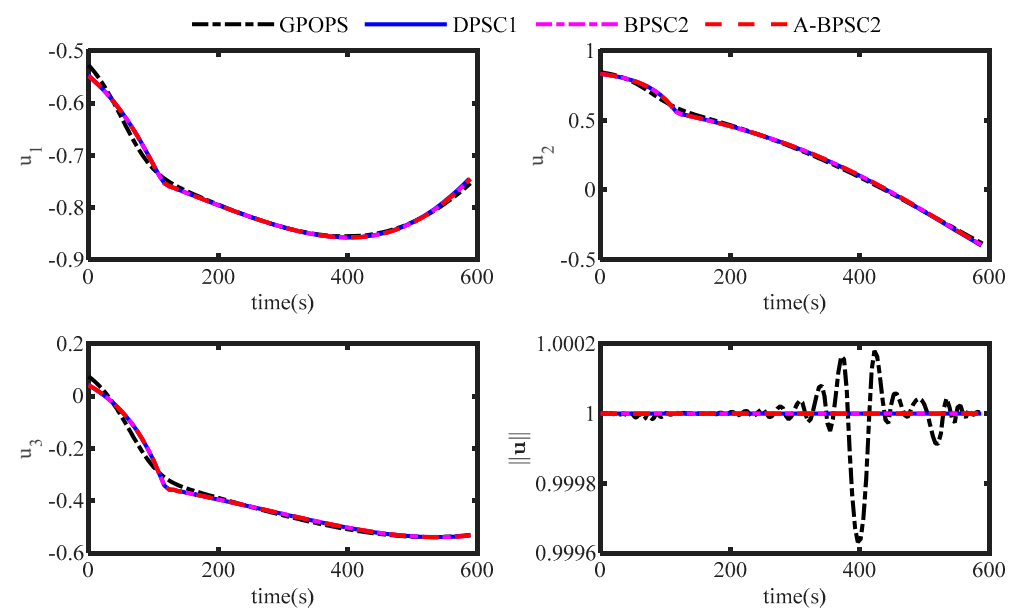


Figure 5. Optimized control input and its 2-norm.

In Figure 6, the profiles of the local flight path angles are presented. The local flight path angle is defined as $\theta_H = \arcsin(\mathbf{R}^T \mathbf{V} / \|\mathbf{V}\|)$ according to [31]. In this work, the flight path angle θ_H is not a direct constraint, and it is constrained by the prescribed R_{2f} .

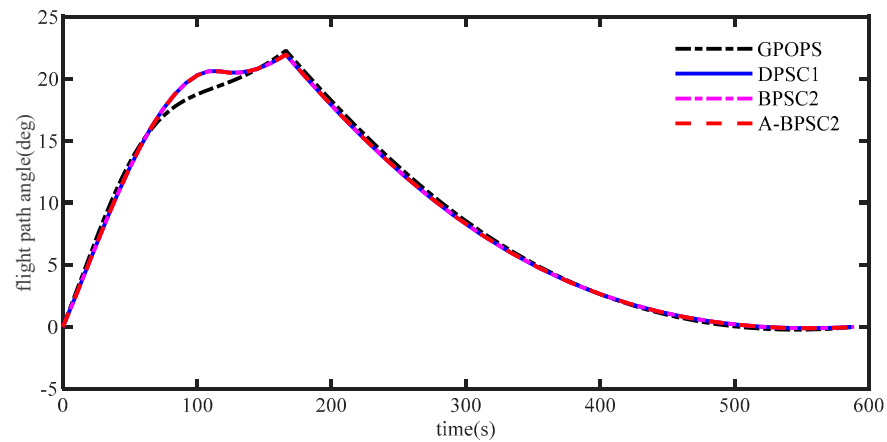


Figure 6. Flight path angle profile.

The iterative procedure of the three SC algorithms is shown in Figure 7, in which the normalized height profiles are presented. The blue dotted lines $h^{(0)}$ resulted from the linear interpolation of three points r_{10} , r_{1f} , and r_{2f} in the sub-procedure 1 of SC algorithms, and the height profiles $h^{(1)}$ denote the vacuum solution; then, the height profiles $h^{(k)}$ ($k > 1$) denote the iterative solutions coupling with aerodynamic force in the sub-procedure 2. It can be seen that after a few iterations, the optimal trajectories are obtained even though we only provide three points as the initial input. We use the optimized control variables in A-BPSC2 as the input and then integrate the system dynamics in Equation (1) by using the 4-order Runge–Kutta algorithm, thereby providing the integral trajectory of TSLV. The optimized height and velocity profiles of A-BPSC2 and the corresponding integral trajectories are shown in Figure 8, which perfectly verified Proposition 2 in Section 3, i.e., we can use the drag force along the previous trajectory to formulate the next iterative problem. To avoid redundancy, the similar results from other methods are not presented here. The satisfactions of the heat flux constraints are shown in Figure 9, in which the heat flux profiles are calculated based on the integral trajectories that resulted from the optimized control inputs.

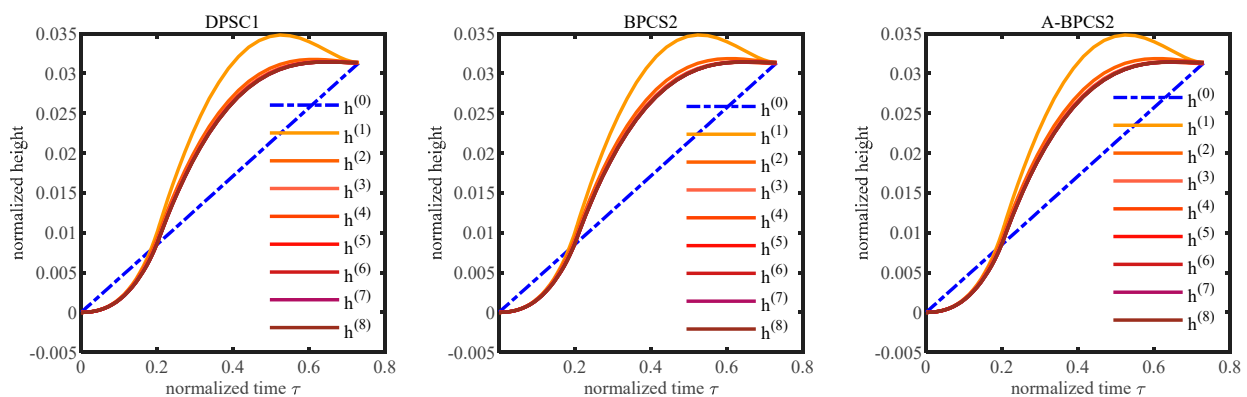


Figure 7. Iterative procedure of SC algorithm.

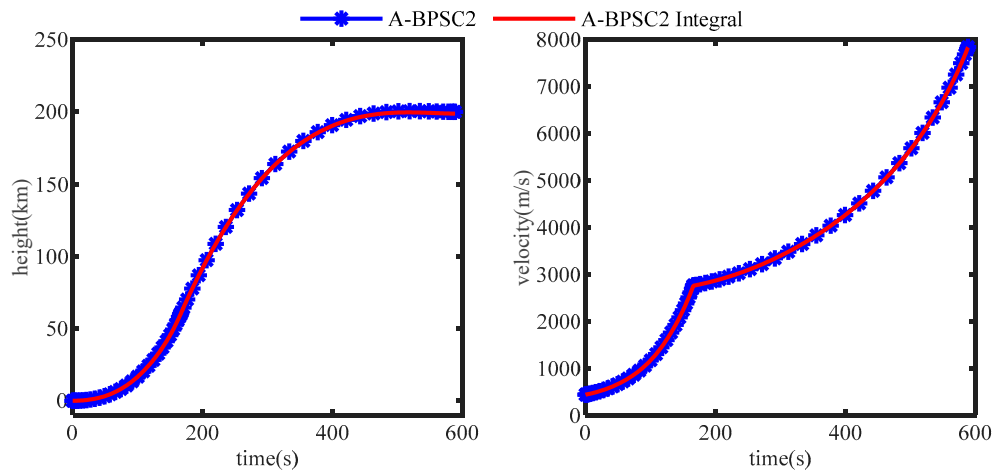


Figure 8. The optimized height and velocity profile versus to the integral results from the optimal control input via the 4-order Runge–Kutta algorithm.

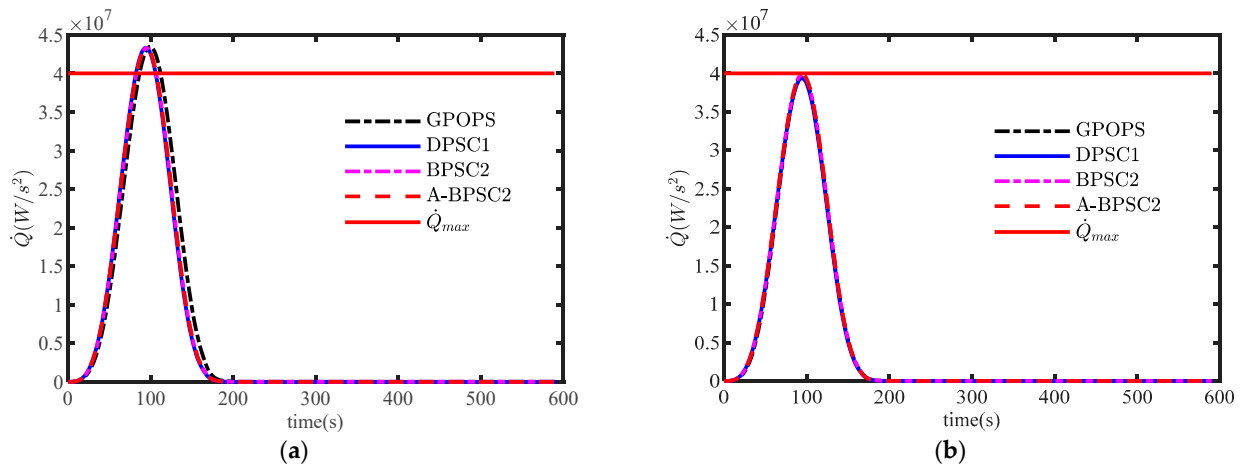


Figure 9. The heat flux history: (a) without heat flux constraint; (b) with heat flux constraints.

Table 2 lists the statistical results of the four mentioned methods. In the numerical experiments, let $N_1 = N_2$; meanwhile, other conditions used in the SC algorithms are completely identical for DPSC1, BPSC2, and A-BPSC2. It can be seen that the objective indices of BPSC2 and A-BPSC2 are identical for the same collocation points; meanwhile, they are smaller than that of DPSC1 and GPOPS. In addition, we calculated the condition numbers of $\mathcal{A}$ and the sparsity ratios of $\mathcal{G}$ in Equation (67) rendered by the three SC algorithms. Here, the sparsity ratio is the ratio of the non-zero elements' number and the total elements' number of the matrix. In the two BPS methods, the increase in the condition number of $\mathcal{A}$ is approximately proportional to N_i , while in the DPSC1, the condition number of $\mathcal{A}$ is much larger and is proportional to the cube of N_i . The well-conditioned $\mathcal{A}$ benefits the subsequent numerical solving [27]; however, the computational cost of BPSC2 has not been reduced, despite the improvement in the condition number of $\mathcal{A}$; on the contrary, the A-BPSC2 reveals the best computational efficiency among the three SC algorithms. From Table 2, the sparsity ratios of $\mathcal{G}$ resulting from BPSC2 exceed 14%, while the sparsity ratios rendered by A-BPSC2 are less than 1%. We henceforth conclude that the main reason for the decrease in computational efficiency for BPSC2 lies in the deteriorated sparsity pattern of $\mathcal{G}$. The results in Table 2 validate the effectiveness of A-BPSC2 for leveraging the sparsity pattern and the number of equality constraints. In addition, the parity of collocation point number N_i also affects the solving time; from our experience, the solution time for choosing an odd N_i is less than that when choosing an even N_i . Compared

with GPOPS, the proposed A-BPSC2 has greatly improved the computational efficiency in the case of the same collocation point number $N_i = 20$.

Table 2. Statistical results of GPOPS, DPSC1, BPSC2, and A-BPSC2.

Methods	N_i	Iterations	Solving Time(s)	t_{2f}^*	Condition of $\mathcal{A}$	Sparsity Ratio of $\mathcal{G}$
DPSC1	20	8	0.3215	587.1776	3.1092×10^5	1.04%
	31	8	0.5203	587.1265	3.5520×10^5	1.00%
	61	8	1.3297	587.1551	1.0888×10^6	0.75%
	91	8	8.8953	587.1466	2.6153×10^7	0.43%
BPSC2	20	8	0.5341	588.9779	1.3843×10^4	14.46%
	31	8	0.6142	588.9623	1.3804×10^4	14.68%
	61	8	2.5494	588.9700	1.3590×10^4	16.27%
	91	8	38.6264	588.9677	1.4905×10^4	18.41%
A-BPCS2	20	8	0.2605	588.9779	8.1242×10^3	0.80%
	31	8	0.2445	588.9623	8.2634×10^3	0.77%
	61	8	0.6263	588.9700	9.5622×10^3	0.58%
	91	8	4.3581	588.9677	1.2722×10^4	0.33%
GPOPS	20	/	6.1670	589.1500	/	

Remark 5. In our numerical experiment, when the equilibration method is closed in ECOS, the DPSC1 cannot converge, but our two BPS methods still can obtain the converged solution. Actually, the equilibration function in ECOS is used to make sure the matrices $\mathcal{A}$ and $\mathcal{G}$ are well conditioned before the application of the interior point method [39]. This reveals that our methods can improve the condition of the resulting SOCP problem, and avoid the equilibrating manipulations in the solver, therefore enhancing the computational efficiency. In Table 2, we enabled the equilibration function in ECOS to obtain the solutions of the three methods. The results demonstrated not only the condition number but also that the sparsity pattern of the resulting SOCP problems dramatically affects the solving efficiency. The proposed A-BPSC2 takes into account both the problem's condition number and sparsity pattern, and is therefore the most outstanding among the three methods.

7. Conclusions

In this paper, a new sequential convex optimization scheme is presented, based on the second-order Birkhoff PS method for efficiently solving the ascent trajectory optimization problem with free final time, taking account of strong nonlinear dynamic constraints as well as the heat flux constraints. By using the second-order BPS method, we can reduce the decision variables as well as the number of equality constraints resulting from the transcription of nonlinear dynamics. However, the introduction of the state-dependent constraints will result in a dramatic increase in the nonzero elements of $\mathcal{G}$ in the SOCP problems, and this seriously affects the solving efficiency. Hence, we originally augment the decision variables in SOCP problems, and combine the advantages of DPSC1 and BPSC2, allowing the slight increase in nonzero elements in $\mathcal{A}$ and the dramatic decrease in the nonzero elements in $\mathcal{G}$. Such a compromise renders an excellent performance in the efficiency of A-BPSC2. In our numerical experiments, since the generalized modeling tool Yalmip is used, the time cost of modeling accounts for almost the majority of the total solution time. Hence, it is necessary to develop a specific modeling tool as well as a solver tightly coupling with the SC algorithm in the future.

Author Contributions: Conceptualization, H.Z. and D.Z.; methodology, D.Z. and H.Z.; software, Y.X. and D.Z.; validation, Q.G., H.Z. and Y.X.; formal analysis, X.X. and Q.G.; investigation, X.X.; resources, Y.X.; data curation, D.Z. and H.Z.; writing—original draft preparation, X.X. and D.Z.; writing—review and editing, D.Z. and H.Z.; visualization, Y.X. and X.X.; supervision, D.Z.; project administration, D.Z.; and funding acquisition, D.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 52472423.

Data Availability Statement: Data are contained within the article.

Acknowledgments: We thank the anonymous reviewers for providing many detailed comments. Their comments greatly improved the quality of this paper.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lu, P.; Pan, B. Highly Constrained Optimal Launch Ascent Guidance. *J. Guid. Control Dyn.* **2010**, *33*, 404–504. [\[CrossRef\]](#)
2. Benedikter, B.; Zavoli, A.; Colasurdo, G.; Pizzurro, S.; Cavallini, E. Convex Optimization of Launch Vehicle Ascent Trajectory with Heat-Flux and Splash-Down Constraints. *J. Spacecr. Rocket.* **2022**, *59*, 900–915. [\[CrossRef\]](#)
3. Lu, P.; Griffin, B.J. Rapid Optimal Multiburn Ascent Planning and Guidance. *J. Guid. Control Dyn.* **2008**, *31*, 9. [\[CrossRef\]](#)
4. Lu, P. Closed-Loop Endoatmospheric Ascent Guidance. *J. Guid. Control Dyn.* **2003**, *26*, 283–294. [\[CrossRef\]](#)
5. Song, E.-J.; Cho, S.; Roh, W.-R. A comparison of iterative explicit guidance algorithms for space launch vehicles. *Adv. Space Res.* **2015**, *55*, 463–476. [\[CrossRef\]](#)
6. Bokanowsky, O.; Bourgeois, E.; Desilles, A. HJB approach for a multi-boost launcher trajectory optimization problem. *IFAC-Pap.* **2016**, *49*, 456–461.
7. Zhao, J.; Li, S. Adaptive mesh refinement method for solving optimal control problems using interpolation error analysis and improved data compression. *J. Frankl. Inst.* **2020**, *357*, 1603–1627. [\[CrossRef\]](#)
8. Trefethen, L.N. *Approximation Theory and Approximation Practice*; SIAM: Philadelphia, PA, USA, 2013.
9. Ross, I.M.; Fahroo, F. Pseudospectral Knotting Methods for Solving Optimal Control Problems. *J. Guid. Control Dyn.* **2004**, *27*, 397–405. [\[CrossRef\]](#)
10. Huntington, G.T. Advancement and Analysis of a Gauss Pseudospectral Transcription for Optimal Control Problems. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2007.
11. Wächter, A.; Biegler, L.T. On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming. *Math. Program.* **2006**, *106*, 25–57. [\[CrossRef\]](#)
12. Gill, P.E.; Murray, W.; Saunders, M.A. *User's Guide for SNOPT Version 7: Software for Large-Scale Nonlinear Programming*; University of California: San Diego, CA, USA, 2008.
13. Blackmore, L.; Açikmeşe, B.; Scharf, D.P. Minimum-Landing-Error Powered-Descent Guidance for Mars Landing Using Convex Optimization. *J. Guid. Control Dyn.* **2010**, *33*, 1161–1171. [\[CrossRef\]](#)
14. Cheng, X.; Li, H.; Zhang, R. Efficient ascent trajectory optimization using convex models based on the Newton–Kantorovich/Pseudospectral approach. *Aerosp. Sci. Technol.* **2017**, *66*, 140–151. [\[CrossRef\]](#)
15. Hu, C.; Bai, X.; Zhang, S.; Yang, H. Successive Convexification for Online Ascent Trajectory Optimization. *IEEE Access* **2021**, *9*, 141843–141860. [\[CrossRef\]](#)
16. Lu, P.; Liu, X.; Yang, R.; Zhang, Z. Ascent Trajectory Optimization with Nonlinearity-Kept Convexification. *IEEE Trans. Aerosp. Electron. Syst.* **2023**, *59*, 3236–3250. [\[CrossRef\]](#)
17. Ma, Y.; Pan, B.; Yan, R. Feasible Sequential Convex Programming with Inexact Restoration for Multistage Ascent Trajectory Optimization. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *59*, 1217–1230. [\[CrossRef\]](#)
18. Wang, Z. A survey on convex optimization for guidance and control of vehicular systems. *Annu. Rev. Control* **2024**, *57*, 100957. [\[CrossRef\]](#)
19. Morelli, A.C.; Hofmann, C.; Toppato, F. Warm Start of Interior-Point Methods Applied to Sequential Convex Programming. *IEEE Trans. Aerosp. Electron. Syst.* **2024**, *60*, 3837–3846. [\[CrossRef\]](#)
20. Adabag, E.; Atal, M.; Gerard, W.; Plancher, B. MPCGPU: Real-Time Nonlinear Model Predictive Control through Preconditioned Conjugate Gradient on the GPU. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024; pp. 9787–9794.

21. Sagliano, M. Generalized hp Pseudospectral Convex Programming for Powered Descent and Landing. In Proceedings of the 2018 AIAA Guidance, Navigation, and Control Conference, Kissimmee, FL, USA, 8–12 January 2018.
22. Zhang, T.; Su, H.; Gong, C. hp-Adaptive RPD based sequential convex programming for reentry trajectory optimization. *Aerosp. Sci. Technol.* **2022**, *130*, 107887. [[CrossRef](#)]
23. Gao, D.; Gong, Y.; Li, C.; Guo, Y.; Fadda, E.; Brandimarte, P. Adaptive pseudospectral successive convex optimization for six-degree-of-freedom powered descent guidance. *Aerosp. Sci. Technol.* **2024**, *155*, 109544. [[CrossRef](#)]
24. Li, D.; Li, Y.; Liu, X.; Yang, B.; Huang, X.; Yang, Y.; Wang, B.; Li, S. Pseudospectral Convex Programming for Free-Floating Space Manipulator Path Planning. *Space Sci. Technol.* **2023**, *3*, 30. [[CrossRef](#)]
25. Ross, I.M.; Proulx, R.J. Further Results on Fast Birkhoff Pseudospectral Optimal Control Programming. *J. Guid. Control Dyn.* **2019**, *42*, 2086–2092. [[CrossRef](#)]
26. Ma, Y.; Pan, B.; Hao, C.; Tang, S. Improved sequential convex programming using modified Chebyshev–Picard iteration for ascent trajectory optimization. *Aerosp. Sci. Technol.* **2022**, *120*, 107234. [[CrossRef](#)]
27. Zhao, D.; Zhang, Z.; Gui, M. Birkhoff Pseudospectral Method and Convex Programming for Trajectory Optimization. In *Autonomous Trajectory Planning and Guidance Control for Launch Vehicles*; Song, Z., Zhao, D., Theil, S., Eds.; Springer Nature: Singapore, 2023; pp. 99–127.
28. Zhang, Z.; Zhao, D.; Li, X.; Kong, C.; Su, M. Convex Optimization for Rendezvous and Proximity Operation via Birkhoff Pseudospectral Method. *Aerospace* **2022**, *2022*, 505. [[CrossRef](#)]
29. Shahmirzaee Jeshvaghani, S.; Novinzaddeh, A.B.; Pazooki, F. Multiple stage satellite launch vehicle ascent optimization using Chebyshev wavelets. *Aerosp. Sci. Technol.* **2015**, *46*, 321–330. [[CrossRef](#)]
30. Benson, D. A Gauss Pseudospectral Transcription for Optimal Control. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2005.
31. Chobotov, V.A. (Ed.) *Orbital Mechanics*; AIAA, Inc.: Reston, VA, USA, 2002.
32. Liu, X.; Shen, Z.; Lu, P. Exact Convex Relaxation for Optimal Flight of Aerodynamically Controlled Missiles. *IEEE Trans. Aerosp. Electron. Syst.* **2016**, *52*, 1181–1192. [[CrossRef](#)]
33. Zhao, D.-J.; Song, Z.-Y. Reentry trajectory optimization with waypoint and no-fly zone constraints using multiphase convex programming. *ACTA Astronaut.* **2017**, *137*, 60–69. [[CrossRef](#)]
34. Benedikter, B.; Zavoli, A.; Colasurdo, G. A Convex Optimization Approach for Finite-Thrust Time-Constrained Cooperative Rendezvous. *Adv. Astronaut. Sci.* **2019**, *171*, 1483–1498.
35. Gath, P.F. CAMTOS—A Software Suite Combining Direct and Indirect Trajectory Optimization Methods; Institute of Flight Mechanics and Flight Control, University of Stuttgart: Stuttgart, Germany, 2002.
36. Liu, X.; Lu, P. Solving Nonconvex Optimal Control Problems by Convex Optimization. *J. Guid. Control Dyn.* **2014**, *37*, 750–765. [[CrossRef](#)]
37. Wang, L.-L.; Samson, M.D.; Zhao, X. A Well-Conditioned Collocation Method Using a Pseudospectral Integration Matrix. *SIAM J. Sci. Comput.* **2014**, *36*, A907–A929. [[CrossRef](#)]
38. Löfberg, J. YALMIP: A Toolbox for Modeling and Optimization in MATLAB2004. In Proceedings of the CACSD Conference, Taipei, Taiwan, 2–4 September 2004.
39. Domahidi, A.; Chu, E.; Boyd, S. ECOS: An SOCP Solver for Embedded Systems. In Proceedings of the European Control Conference, Zurich, Switzerland, 17–19 July 2013.
40. Domahidi, A. *Methods and Tools for Embedded Optimization and Control*; ETH Zurich: Zürich, Switzerland, 2013.
41. Patterson, M.A.; Rao, A.V. 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.* **2014**, *41*, 1–37. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.