aerospace

MDPI

Article

An hp-Legendre Pseudospectral Convex Method for 6-Degree-of-Freedom Powered Landing Problem

Jun Huang and Yidong Zeng *

College of Computer Science and Technology, Southwest University of Science and Technology, Mianyang 621010, China; huangjunswust@foxmail.com
* Correspondence: zydhps@outlook.com

Abstract: This paper presents a fast trajectory optimization method combining the hp-Legendre pseudospectral method and convex optimization for the 6-Degree-of-Freedom rocket-powered landing problem. To accelerate calculations, this paper combines the Legendre pseudospectral method with a linearization method for convexification, and an hp method that can divide the mesh is introduced to reduce the computational workload. In terms of accuracy, a trust region update strategy that can control the solution process is presented to approximate the original problem iteratively. Convergence analysis is provided as evidence, substantiating that any solution produced by the hp-Legendre pseudospectral convex method is not only feasible but potentially optimal for the original problem. The effectiveness of the proposed method is demonstrated by numerical experiments. When compared, the proposed method achieves higher calculation accuracy in solving the 6-Degree-of-Freedom rocket-powered landing trajectory problem, while taking into account rocket attitude control.

Keywords: Legendre pseudospectral method; convex optimization; 6-Degree-of-Freedom powered landing

Citation: Huang, J.; Zeng, Y. An hp-Legendre Pseudospectral Convex Method for 6-Degree-of-Freedom Powered Landing Problem. *Aerospace* **2023**, *10*, 849. https://doi.org/10.3390/aerospace10100849

Academic Editor: Angelo Cervone

Received: 23 July 2023
Revised: 17 September 2023
Accepted: 26 September 2023
Published: 28 September 2023

1. Introduction

Trajectory optimization plays a particularly important role in the design of space aircraft and high-altitude, high-speed, high-maneuverability aircraft. It runs through the entire aircraft design process, affecting the overall aerodynamic layout, guidance and control, power, and the structure of multiple subsystems.

Trajectory optimization of aircraft in the aerospace field is fundamentally an optimal control problem. The methods for solving the optimal control problem are mainly divided into indirect methods and direct methods. Indirect methods require extensive mathematical derivation and are sensitive to initial values, making them unsuitable for solving complex large-scale optimal control problems. Direct methods transcribe an optimal control problem into a nonlinear programming problem (NLP) using a parameterization technique [1]. Due to the development of computers, direct methods have been further developed and applied in many fields. In direct methods, the most commonly applied techniques are pseudospectral methods [1,2] and the convex optimization method [3], which are considered to have the potential to be developed into online optimization methods.

Many references have proven the Karush–Kuhn–Tucker (KKT) conditions of the pseudospectral methods [4]. However, in practice, the pseudospectral method, which is merely a method that discretizes the optimal control problem into nonlinear programming, does not have a direct solution. It needs to be combined with sequential quadratic programming (SQP) to be solved. Although the pseudospectral method can guarantee the KKT of the discretized optimal control problem, it does not prove that there is a solution in a certain amount of time. This factor greatly limits the online application of the pseudospectral method. The solution of nonlinear programming cannot always converge to an optimal result in all cases. The global convergence of the convex optimization method is considered an important factor in accelerating the solution efficiency.

A natural idea is to combine the pseudospectral method (PM) [5] and convex optimization to break through the traditional PM. This can be called the pseudospectral convex method (PCM). The challenge of rocket-powered landing is a prominent area of research within the field of trajectory optimization, and the application of PCM to achieve faster and more precise landing trajectory solutions has emerged as a focal point of research. Sagliano [6] combined the flipped Radau pseudospectral method with the convex optimization method to tackle the Mars powered descent problem, The problem involves a 3-Degree-of-Freedom (3-DOF) model that considers both position and velocity in three-dimensional space. Wang [7] introduced the pseudospectral improved successive convexification (P-iSC) algorithm for 3-DOF rocket landing problems, assuming ideal rocket attitude control. In various trajectory optimization problems, PCMs are widely employed. Li [8] proposed the successive Chebyshev pseudospectral convex optimization method and successfully addressed spacecraft orbit transfer problems. Yu [9] developed an innovative convex optimization algorithm based on the Chebyshev pseudospectral method, effectively solving reentry trajectory optimization problems. Zhang [10] introduced a novel method that combines convex techniques with Birkhoff pseudospectral methods to address rendezvous and proximity operation problems while considering merely nonlinear inequality constraints.

The characteristic of the pseudospectral method is that increasing the number of collocations will increase the accuracy of the solution. However, the increase in the number of points will also increase the calculation burden for optimization. There is a contradiction between the number of discrete allocation points and the computing speed. The idea of using hp mesh division for sparse algebraic equations in the pseudospectral method is borrowed in PCM to alleviate conflicts. In hp mesh division, h represents the total number of meshes, and p represents the number of collocations in a mesh [11]. Sagliano [12] introduced a generalized hp pseudospectral method to solve a 3-DOF landing problem. Lei [13] proposed a convex optimization method based on an hp-adaptive pseudospectral method for planning the trajectory of a 3-DOF powered landing.

In summary, the integration of the pseudospectral method and convex optimization has emerged as an innovative approach [14] for addressing numerous optimal control problems. In the context of rocket landing trajectory problems, extensive research has been conducted on the 3-DOF optimal control problem. However, it is worth noting that the 3-DOF model assumes ideal attitude control, which introduces a disparity between the idealized model and the real-world scenario.

In this paper, we focus on addressing the 6-Degree-of-Freedom rocket landing problem, considering attitude variation as a research objective. We introduce a novel hp-Legendre pseudospectral convex method to tackle the nonlinear optimal control problem associated with 6-Degree-of-Freedom powered landing. The method is built upon a foundation of the Legendre pseudospectral method and convex optimization techniques. We employ hp mesh division technology to reduce the computational burden of algebraic equations. Recognizing the complexity of achieving an exact solution in a single calculation, we incorporate a trust region update strategy to control the iterative approximation solution process.

The remaining paper is structured as follows: Section 2 presents the 6-Degree-of-Freedom powered landing problem with relevant constraints. In Section 3, we propose the hp-Legendre pseudospectral convex method (LPCM). Section 4 introduces the hp-Legendre pseudospectral convex method with trust region strategy (hp-LPCM-TRS) along with the accompanying convergence analysis. In Section 5, we showcase numerical simulation of the 6-Degree-of-Freedom rocket-powered landing problem to illustrate the effectiveness and performance of the proposed method. Section 6 serves as the conclusion of the article.

2. Six-DOF Rocket-Powered Landing Problem

To provide a clear understanding of the problem, we will elaborate on the 6-Degree-of-Freedom powered landing problem [15]. The dynamic equations, boundary constraints, and path constraints are presented below.

The mass depletion dynamics are described as follows:

$$\dot{m} = -\frac{||U||_2}{I_{sp} g_0} \tag{1}$$

where the vacuum-specific-impulse I_{sp} and Earth's standard gravity constant g_0 are defined. The thrust vector is denoted as $U = \left[T_x, T_y, T_z\right]^T$. The position vector and velocity vector are denoted as $R = \left[r_x, r_y, r_z\right]^T$ and $V = \left[v_x, v_y, v_z\right]^T$, respectively. The differential equations for position and velocity are as follows:

$$\dot{R} = V \tag{2}$$

$$\dot{V} = \frac{1}{m} C_B U + G \tag{3}$$

where $G = [0, 0, g_0]^T$, and the matrix C, which encodes the attitude transformation, is denoted as:

$$C_B = \begin{bmatrix} -2q_3^2 - 2q_4^2 + 1 & 2q_1q_4 + 2q_2q_3 & -2q_1q_3 + 2q_2q_4 \\ -2q_1q_4 + 2q_2q_3 & -2q_2^2 - 2q_4^2 + 1 & 2q_1q_2 + 2q_3q_4 \\ 2q_1q_3 + 2q_2q_4 & -2q_1q_2 + 2q_3q_4 & -2q_2^2 - 2q_3^2 + 1 \end{bmatrix} \tag{4}$$

It consists of an attitude quaternion $Q = [q_1, q_2, q_3, q_4]^T$. Quaternions do not have singularities, and, in contrast to other representations such as Euler angles, they do not suffer from issues like Gimbal lock. In addition, quaternions are suitable for computer computation. More details about C can be found in reference. The attitude quaternion differential equation is as follows:

$$\dot{Q} = \frac{1}{2} O_w Q \tag{5}$$

where the skew-symmetric matrix O_w is denoted as follows:

$$O_w = \begin{bmatrix} 0 & -w_x & -w_y & -w_z \\ w_x & 0 & w_z & -w_y \\ w_y & -w_z & 0 & w_x \\ w_z & w_y & -w_x & 0 \end{bmatrix} \tag{6}$$

where w_x, w_y, and w_z denote the angular velocity of the rocket rotating around the three axes in the arrow body coordinate system, respectively. Finally, the three differential equations for the angular velocity of the rotation are as follows:

$$\dot{W} = M - S_w W \tag{7}$$

where M is a torque matrix and S_w is a skew-symmetric matrix, as denoted by

$$S_w = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix} \tag{8}$$

$$M = J_B^{-1} R_B U \tag{9}$$

where J_B represents the inertial tensor of the vehicle, and R_B represents the constant position vector, as denoted by

$$J_B = \begin{bmatrix} j_{b1} & 0 & 0 \\ 0 & j_{b2} & 0 \\ 0 & 0 & j_{b3} \end{bmatrix} \tag{10}$$

$$R_B = \begin{bmatrix} 0 & -r_{b3} & r_{b2} \\ r_{b3} & 0 & -r_{b1} \\ -r_{b2} & r_{b1} & 0 \end{bmatrix} \tag{11}$$

where $r_b = \left[r_{b_1}, r_{b_2}, r_{b_3}\right]$ represents the gimbal-point position vector and $j_b = [j_{b1}, j_{b2}, j_{b3}]$ represents the inertial vector of the vehicle.

The state variable vector $X = [m, R, V, Q, W]^T$ is defined, and the control variable vector $U = \left[T_x, T_y, T_z\right]^T$ is also defined. They are functions of time. The 6-Degree-of-Freedom powered landing problem must satisfy boundary conditions and path constraints.

The initial and terminal boundary variables are

$$X_{t_0} = [m(t_0), R(t_0), V(t_0), Q(t_0), W(t_0)] \tag{12}$$

$$X_{t_f} = \left[m\left(t_f\right), R\left(t_f\right), V\left(t_f\right), Q\left(t_f\right), W\left(t_f\right)\right] \tag{13}$$

The given initial and terminal conditions values are provided by

$$X_0 = [m_0, R_0, V_0, Q_0, W_0] \tag{14}$$

$$X_f = \left[m_f, R_f, V_f, Q_f, W_f\right] \tag{15}$$

The boundary conditions are as follows:

$$X_{t_0} = X_0 \tag{16}$$

$$X_{t_f} = X_f \tag{17}$$

where the subscripts t_0 and t_f indicate the state variables at the initial time and at the terminal time, respectively.

The path constraints are given as follows. The mass of the rocket is restricted by the following constraint:

$$m_{dry} \le m(t) \le m_{wet} \tag{18}$$

where m_{dry} is the structural mass and m_{wet} is m_{dry} plus fuel mass. The trajectory is restricted within an upward-facing glide-slope cone with an angle γ in the range of $[0^\circ, 90^\circ]$.

$$\sqrt{r_x(t)^2 + r_y(t)^2} \le \frac{r_z(t)^2}{tan(\gamma)} \tag{19}$$

To avoid excessive tilt angles θ in the trajectory, it is limited by a maximum value of θ_{max}. The path constraint for θ_{max} is denoted as follows:

$$\sqrt{q_2(t)^2 + q_3(t)^2} \le \sqrt{\frac{1 - cos(\theta_{max})}{2}} \tag{20}$$

Furthermore, a maximum angular rate of w_{max} is used to limit W by

$$||W(t)||_2 \le w_{max} \tag{21}$$

For convenience, $W(t) = \left[w_x(t), w_y(t), w_z(t)\right]$ is a vector of angular rate functions. The commanded thrust vector is constrained within the magnitude interval $[T_{min}, T_{max}]$ and the gimbal angle interval $[0, \delta_{max}]$, where $\delta_{max} \in (0°, 90°)$.

$$0 < T_{min} \leq ||U(t)||_2 \leq T_{max} \tag{22}$$

$$cos(\delta_{max})||U(t)||_2 \leq T_z(t) \tag{23}$$

where the thrust magnitude lower bound constraint is non-convex, it needs to be linearized using a first-order Taylor series approximation. The linearized lower bound constraint is obtained by

$$T_{min} \leq \frac{\hat{U(t)}^T}{\left|\left|\hat{U(t)}\right|\right|_2} U(t) \tag{24}$$

$\hat{U(t)}$ is a guessed value. The maximum mass at the terminal time is set as the performance index. In summary, we have the following continuous 6-Degree-of-Freedom powered landing optimal control problem, denoted as P_L:

$$(P_L)\begin{cases} Min \circ J = -m\left(t_f\right) \\ s.t \circ \dot{m(t)} = -\frac{||U(t)||_2}{I_{sp}g_0} \dot{R(t)} = V(t) \dot{V(t)} = \frac{1}{m} C_B U(t) + G \\ \dot{Q(t)} = \frac{1}{2} O_w Q(t) \dot{W(t)} = M - S_w W(t) \\ X_{t_0} = X_0 X_{t_f} = X_f m_{dry} \leq m(t) \leq m_{wet} \\ \sqrt{r_x(t)^2 + r_y(t)^2} \leq \frac{r_z(t)^2}{tan(\gamma)} \sqrt{q_2(t)^2 + q_3(t)^2} \leq \sqrt{\frac{1-cos(\theta_{max})}{2}} \\ ||W(t)||_2 \leq w_{max} \\ ||U(t)||_2 \leq T_{max} \\ T_{min} \leq \frac{\hat{U(t)}^T}{||\hat{U(t)}||_2} U(t)\ cos(\delta_{max})||U(t)||_2 \leq T_z(t) \end{cases} \tag{25}$$

All path constraints of problem P_L have been converted to convex constraints. However, due to the nonlinear nature of the dynamic equations, the entire problem remains non-convex. The next section will demonstrate how to employ the pseudospectral convex method to linearize the nonlinear dynamic equations within the problem step by step. The transformation will ultimately render problem P_L as a convex problem.

3. hp-Legendre Pseudospectral Convex Method

3.1. Combining Legendre Pseudospectral Method and Convex Optimization

The Legendre pseudospectral convex method (LPCM) principle involves discretizing the state at Legendre–Gauss–Lobatto (LGL) collocation points, multiplying it with a differential matrix to obtain the corresponding differential values, and subsequently completing the algebraic substitution of the differential components. This process can be broken down into the following steps:

In accordance with the principle of the Legendre pseudospectral method, it is essential to map the real time interval $t \in \left[t_0,\ t_f\right]$ to $\tau \in [-1,\ 1]$. During this mapping, the state variable function $x(\tau)$ and the control variable function $u(\tau)$ are approximated using Lagrange interpolating polynomials.

$$x(\tau) = \sum_{i=0}^{N} L_i(\tau) X_i \tag{26}$$

$$u(\tau) = \sum_{i=0}^{N} L_i(\tau) U_i \tag{27}$$

$L_i(\tau)$ is the Lagrange interpolation basis function and can be calculated by

$$L_i(\tau) = \prod_{j=0, j\neq i}^{N} \frac{\tau - \tau_j}{\tau_i - \tau_j} \tag{28}$$

The derivation of $x(\tau)$ and $u(\tau)$ with respect to time τ is as follows:

$$\dot{x}(\tau_k) = \sum_{i=0}^{N} \dot{L}_i(\tau_k) X_i = \sum_{i=0}^{N} D_{ki} X_i \tag{29}$$

$$\dot{u}(\tau_k) = \sum_{i=0}^{N} \dot{L}_i(\tau_k) U_i = \sum_{i=0}^{N} D_{ki} X_i \tag{30}$$

where $k = 0, 1, \ldots, N; D^{(N+1)\times(N+1)}$ is a differential matrix. The differential value, which is of the Lagrange basis function at each τ, has a clear mathematical expression as follows:

$$(D_{ki}) \begin{cases} \frac{P_N(\tau_k)}{P_N(\tau_i)(\tau_k - \tau_i)}, \ i \neq k \\ -N(N+1)/4, \ i = k = 0 \\ N(N+1/4), \ i = k = N \\ 0, \ otherwise \end{cases} \tag{31}$$

In the above expression, $P_{N(\tau_i)}$ is the Legendre orthogonal polynomial. For detailed information, please refer to reference [15]. Integral terms in J need to be converted to matrix multiplication using a quasi-Gaussian quadrature formula.

$$\int_{t_0}^{t_f} G(x(t), u(t), t) dt = \sum_{k=0}^{N} G_k w_k \tag{32}$$

Here, w_j is the Legendre integral weight, which is mathematically expressed as follows:

$$w_j = \frac{2}{N(N+1)} \frac{1}{P_N^2(x_j)}, 0 \leq j \leq N \tag{33}$$

The discretized problem, denoted as P_N, is established based on the aforementioned steps:

$$(P_N) \begin{cases} Min \qquad J = E(x(-1), x(1)) + \sum\limits_{k=0}^{N} G_k w_k \\ \qquad s.t \quad \sum\limits_{i=0}^{N} D_{ki} X = f(X_k, U_k, \tau_k) \\ \qquad C_1(X_i, U_i) = 0, \ \ i = 0, 1, \ldots, N \\ \qquad C_2(X_i, U_i) \leq 0, \ i = 0, 1, \ldots, N \end{cases} \tag{34}$$

Now, C_1 and C_2 represent the discretization formula of continuous path constraints. J represents the discretization formula of the continuous performance index. Continuous dynamics have been replaced by algebraic matrix multiplication. According to problem P_N, the pseudospectral discretized powered landing problem is given by

$$
(P_{NL})\begin{cases}
Min \circ J = -m_N \\
s.t \circ \sum_{i=0}^{N} D_{ki}m_k = -\frac{||U_k||_2}{I_{sp}g_0} \sum_{i=0}^{N} D_{ki}R_k = V_k \sum_{i=0}^{N} D_{ki}V_k = \frac{1}{m}C_{BU_k} + G \\
\sum_{i=0}^{N} D_{ki}Q_k = \frac{1}{2}O_w Q_k \sum_{i=0}^{N} D_{ki}W_k = M - S_w W_k \\
X_{t_0} = X_0 \ X_{t_f} = X_N \ m_{dry} \le m_k \le m_{wet} \\
\sqrt{r_{x_k}^2 + r_{y_k}^2} \le \frac{r_{z_k}^2}{tan(\gamma)} \sqrt{q_{2_k}^2 + q_{3_k}^2} \le \sqrt{\frac{1-cos(\theta_{max})}{2}} \\
||W_k||_2 \le w_{max} \\
||U_k||_2 \le T_{max} \\
T_{min} \le \frac{\hat{U}_k^T}{||\hat{U}_k||_2}U_k \ \ cos(\delta_{max})||U_k||_2 \le T_{z_k}, k = 0,1,2,..,N
\end{cases} \tag{35}
$$

The subscript k denotes the k-th discretization value. For example, m_k represents k-th variable derived from a series of discrete mass variables. The above rocket landing trajectory problem is a nonlinear programming problem, especially because the path constraint is convex, but the equality constraint is nonlinear.

The optimal control problem is converted into a nonlinear programming problem using the Legendre pseudospectral method; while it can be solved with SQP, the solution speed cannot be guaranteed. In this section, we transform the nonlinear programming problem into a second-order cone problem (*SOCP*) in convex optimization, which can be efficiently solved. The basic form of the *SOCP* is as follows:

$$
(P_{SOCP})\begin{cases}
min \circ a_i^T x \\
s.t. \circ A_0 x = b_0 \\
||A_i x + b_i|| \le c_i^T + d, i = 1,2,..,m
\end{cases} \tag{36}
$$

The symbol $||\cdot||$ represents the 2-norm. In the *SOCP*, some linear equality constraints are required, and the feasible region for inequality constraints is second-order conical. The Legendre pseudospectral convex optimization method is employed to convert a nonlinear optimal control problem into the *SOCP*. However, the transformed problem may not be solved initially, and the solvability of the problem needs to be improved by relaxation and trust region methods. The process is as follows:

Linearizing the nonlinear terms. In problem P_N, the dynamics, which often contain nonlinear terms, need to be linearized. Here the nonlinear dynamics are linearized using the first-order Taylor formula.

$$
f(x) = f(\hat{x}) + f'(\hat{x})(x - \hat{x}) + o(x - \hat{x}) \tag{37}
$$

where $f(x)$ is a nonlinear function, $\hat{x}$ is a known quantity, $f'(x)$ is the first derivative, and o represents an infinitesimal of higher order.

In dynamics, the first-order differential equation is written as follows:

$$
\dot{x} = \frac{dx}{d\tau} = f(x,u) \tag{38}
$$

The right-hand side is a nonlinear function, and the left-hand side is a derivative of the state variable. Taylor expansion can be used to linearize the right-hand side. The linearization equation is as follows:

$$
f(x,u)|_{(\hat{x},\hat{u})} = A(\hat{x},\hat{u}) \cdot (x - \hat{x}) + B(\hat{x},\hat{u}) \cdot (u - \hat{u}) + f(\hat{x},\hat{u}) \tag{39}
$$

In the above equation, $\hat{x}$ and $\hat{u}$ represent the last generation state variable and control variable, respectively. x and u are unknown variables that need to be solved. A represents the partial derivative of $f(x,u)$ with respect to state variable x, and B represents the partial derivative of $f(x,u)$ with respect to control variable u. They can be written as follows:

$$A = f_x = \frac{\partial f(x,u)}{\partial x} \tag{40}$$

$$B = f_u = \frac{\partial f(x,u)}{\partial u} \tag{41}$$

$\dot{x}$ can be represented as follows:

$$\dot{x} = \frac{dx}{d\tau} = f(x,u) = L(x,u) + R(\hat{x},\hat{u}) \tag{42}$$

L represents a linear system:

$$L(x,u) = Ax + Bu \tag{43}$$

R represents the difference between the actual dynamic value of last generation solution and the linearized system value.

$$R(\hat{x},,\hat{u}) = f(\hat{x},,\hat{u}) - (A\hat{x} + B\hat{u}) \tag{44}$$

For $DX = \dot{X}$, the algebraic equation can be written as follows:

$$\sum_{i=0}^{N} D_{ki}X_i = L(X_k,U_k) + R\left(\hat{X}_k,\hat{U}_k\right) \tag{45}$$

Involving relaxation and trust regions. A relaxation variable, denoted as v_f, is introduced to manage the feasibility of relaxations [7]. Additionally, we have δ_x and δ_u, representing trust regions for state and control variables, respectively. These trust regions are crucial in ensuring that the solution remains in proximity to the linearization point, thereby enhancing numerical stability throughout the solution process [14].

$$\sum_{i=0}^{N} D_{ki}X_i - \left(L(X_i,U_i) + R\left(\hat{X}_i,\hat{U}_i\right)\right) = v_f \tag{46}$$

$$\left|X - \hat{X}\right| \le \delta_x \tag{47}$$

$$\left|U - \hat{U}\right| \le \delta_u \tag{48}$$

The Legendre pseudospectral convex problem, denoted by P_{Nc}, is generally represented in the following form:

$$(P_c)\begin{cases} Min \circ J_c[X,U] = J + w_v \cdot \left|\left|v_f\right|\right| \\ s.t \circ \sum\limits_{i=0}^{N} D_{ki}X_i - \left(L(X_i,U_i) + R\left(\hat{X}_i,\hat{U}_i\right)\right) = v_f \\ C_1(X_i,U_i,\tau_i) = 0,\ i = 0,1,\ldots,N \\ C_2(X_i,U_i,\tau_i) \le 0,\ i = 0,1,\ldots,N \\ \left|X - \hat{X}\right| \le \delta_x \\ \left|U - \hat{U}\right| \le \delta_u \\ j = 0,1,\ldots,N \end{cases} \tag{49}$$

Solving problem. Based on P_{Nc}, the problem P_{NC} is simplified and denoted as follows:

$$(P_{NLC})\begin{cases} Min \circ J = -m_N \\ s.t \sum_{i=0}^{N} D_{ki} m_k = \left(-\frac{||\hat{U}_k||_2}{I_{sp} g_0} + A_m (X_k - \hat{X}_k) + B_m (U_k - \hat{U}_k) \right) \\ \sum_{i=0}^{N} D_{ki} R_k = (\hat{V}_k + A_R (X_k - \hat{X}_k) + B_R (U_k - \hat{U}_k)) \\ \sum_{i=0}^{N} D_{ki} V_k = \left(\frac{1}{m} C \hat{U}_k + G + A_V (X_k - \hat{X}_k) + B_V (U_k - \hat{U}_k) \right) \\ \sum_{i=0}^{N} D_{ki} Q_k = \left(\frac{1}{2} O_w \hat{Q}_k + A_Q (X_k - \hat{X}_k) + B_Q (U_k - \hat{U}_k) \right) \\ \sum_{i=0}^{N} D_{ki} W_k = \left(J_B^{-1} R_B \hat{U}_k - S_w \hat{W}_k + A_W (X_k - \hat{X}_k) + B_W (U_k - \hat{U}_k) \right) \\ X_{t_0} = X_0 X_{t_f} = X_N \\ m_{dry} \le m_k \le m_{wet} \\ ||W_k||_2 \le w_{max} \ ||U_k||_2 \le T_{max} \\ \sqrt{r_{x_k}^2 + r_{y_k}^2} \le \frac{r_{z_k}^2}{tan(\gamma)} \sqrt{q_{2_k}^2 + q_{3_k}^2} \le \sqrt{\frac{1 - cos(\theta_{max})}{2}} \\ T_{min} \le \frac{\hat{U}_k^T}{||\hat{U}_k||_2} U_k \, cos(\delta_{max}) ||U_k||_2 \le T_{z_k} \\ k = 0, 1, 2, .., N \end{cases} \tag{50}$$

where A_m, A_R, A_Q, A_V, and A_W represent the derivatives of each dynamic with respect to state variable X, respectively, and B_m, B_R, B_Q, B_V, and B_W represent the derivatives of dynamics with respect to control variable U. Further details can be found in Appendix A.

The problem P_{NLC} is an SOCP that can be solved using convex optimization algorithms. The solution obtained from P_{NLC} represents the global convergence solution of the problem, but it should be noted that this solution is a relaxation of the original problem. A real solution to the original problem is achieved only when the relaxation equals zero.

Equations (26) to (50) represent the mathematical formulation of the Legendre pseudospectral convex method for solving the 6-Degree-of-Freedom powered landing problem.

3.2. Mesh Division Using hp Method

After obtaining the relaxed SOCP form of the optimal control problem in the previous section, the solution scope of the problem is significantly expanded. However, the solution speed is related to the scale of the established SOCP [3]. Using hp mesh division, the large SOCP can be divided into multiple sub-SOCPs connected by boundary conditions. The solution of the established relaxed SOCP is constrained by a trust region to remain close to the linearization point, which may not always yield the optimal solution. By adjusting the size of the trust region, the solution space of the SOCP can be gradually changed, allowing it to converge towards the solution of the original problem.

The hp-Legendre pseudospectral convex method (hp-LPCM) is proposed in this section. We will explain how this method further subdivides the relaxed SOCP and updates the trust region to bring the relaxed solution closer to the solution of the original problem during the iterative solution process. The specific steps are as follows:

Determine the total number of collocations that can be assigned to the problem P_{NLC}, and the total number of meshes, denoted as H. Different numbers of collocations can be assigned to different meshes, denoted by $N_{hp} = [N^0, N^1, N^2, \ldots, N^H]$. These numbers must satisfy the following constraint:

$$p = \sum_{i=0}^{H} N^i \tag{51}$$

$D = \left[D^0, D^1, D^2, .., D^h\right]$ is calculated according to N_{hp}. The total time interval in each segment is represented by $T_{hp} = \left[\left(t_0^0, t_f^0\right), \left(t_0^1, t_f^1\right), \ldots, \left(t_0^H, t_f^H\right)\right]$.

$$\sum_{i=0}^{N} D^j{}_{ki} X_i^j = \frac{t_f^j - t_0^j}{2} f\left(X_i^j, U_i^j\right), j = 0, 1, 2, .., H \tag{52}$$

where X^j and U^j represent the state vector and control vector of j-th mesh, respectively. The time in T must satisfy the following constraints, for relating the sub-mesh all together.

$$t_f^j = t_0^{j+1}, j = 0, 1, 2, \ldots, H-1 \tag{53}$$

$$X_0^j = X_f^{j+1}, , j = 0, 1, 2, \ldots, H-1 \tag{54}$$

It is obvious that control variables are not correlated because control variables might appear as jump points in many cases.

Linearizing each mesh using the linearization method in LPCM. The specific equations for this process are as follows:

$$\sum_{i=0}^{N} D^j{}_{ki} X_i^j = \frac{t_f^j - t_0^j}{2}\left(L\left(X_i^j, U_i^j\right) + R\left(\hat{X}_i^j, \hat{U}_i^j\right)\right), j = 0, 1, 2, .., h \tag{55}$$

where $\hat{X}_i^j$ and $\hat{U}_i^j$ represent the state vector and control vector of j-th mesh in the last generation.

Relaxing each mesh using the relaxation technique in LPCM and adding trust regions δ_x^j and δ_u^j to limit the variation of the state and control quantities around the linear point.

$$\sum_{i=0}^{N} D^j{}_{ki} X_i^j - \frac{t_f^j - t_0^j}{2}\left(L\left(X_i^j, U_i^j\right) + R\left(\hat{X}_i^j, \hat{U}_i^j\right)\right) = v_f, j = 0, 1, 2, .., h \tag{56}$$

$$\left|X^j - \hat{X}^j\right| \leq \delta^j{}_x \ , \ \ j = 0, 1, 2, \ldots, h \tag{57}$$

$$\left|U^j - \hat{U}^j\right| \leq \delta^j{}_u \ , \ \ j = 0, 1, 2, \ldots, h \tag{58}$$

The hp-LPCM formulas for problem P_{hpc} are summarized as follows:

$$\left(P_{hpc}\right)\begin{cases} Min \circ J_{hpc}[X, U] = -m_p + w_v \cdot \left|\left|v_f\right|\right|_2 \\ s.t \circ \sum\limits_{i=0}^{N} D^j{}_{ki} X_i^j - \frac{t_f^j - t_0^j}{2}\left(L\left(X_i^j, U_i^j\right) + R\left(\hat{X}_i^j, \hat{U}_i^j\right)\right) = v_f, \\ t_f^j = t_0^{j+1}, \ j = 0, 1, 2, \ldots, H-1 \\ X_0^j = X_f^{j+1}, \ j = 0, 1, 2, \ldots, H-1 \\ C_1\left(X_i^j, U_i^j\right) = 0 \\ C_2\left(X_i^j, U_i^j\right) \leq 0 \\ p = \sum\limits_{j=0}^{h} N^j \\ \left|X - \hat{X}\right| \leq \delta_x \\ \left|U - \hat{U}\right| \leq \delta_u \\ j = 0, 1, 2, .., h-1 \\ i = 0, 1, \ldots, N \end{cases} \tag{59}$$

The relaxed SOCP problem for rocket landing is processed through a series of steps to obtain the mesh division relaxed SOCP problem, denoted as P_{NLC}^{hp}.

$$
\left(P_{NLC}^{hp}\right)\begin{cases}
Min \circ J = -m_N + w_v \cdot \left|\left|v_f\right|\right|_2 \\
s.t \sum_{i=0}^{N^j} D_{ki}^j m_k^j = \sigma^j \left(-\frac{\left|\left|\hat{U}_k^j\right|\right|_2}{I_{sp} g_0} + A_m \left(X_k^j - \hat{X}_k^j \right) + B_m \left(U_k^j - \hat{U}_k^j \right) \right) + v_f \\
\sum_{i=0}^{N^j} D_{ki}^j R_k^j = \sigma^j \left(\hat{V}_k^j + A_R \left(X_k^j - \hat{X}_k^j \right) + B_R \left(U_k^j - \hat{U}_k^j \right) \right) + v_f \\
\sum_{i=0}^{N^j} D_{ki}^j V_k^j = \sigma^j \left(\frac{1}{m} C \hat{U}_k^j + G + A_V \left(X_k^j - \hat{X}_k^j \right) + B_V \left(U_k^j - \hat{U}_k^j \right) \right) + v_f \\
\sum_{i=0}^{N^j} D_{ki}^j Q_k^j = \sigma^j \left(\frac{1}{2} O_w \hat{Q}_k^j + A_Q \left(X_k^j - \hat{X}_k^j \right) + B_Q \left(U_k^j - \hat{U}_k^j \right) \right) + v_f \\
\sum_{i=0}^{N^j} D_{ki} W_k^j = \sigma^j \left(J_B^{-1} R_B \hat{U}_k^j - S_w \hat{W}_k^j + A_W \left(X_k^j - \hat{X}_k^j \right) + B_W \left(U_k^j - \hat{U}_k^j \right) \right) + v_f \\
X_{t_0} = X_0 \; X_{t_f} = X_{N^{H-1}} \;,\; k = 0,1,2,\ldots,N^j, j = 0,1,2,\ldots,H-1 \\
m_{dry} \le m_i \le m_{wet} \\
\sqrt{r_{x_i}^2 + r_{y_i}^2} \le \frac{r_{z_i}^2}{tan(\gamma)} \\
\sqrt{q_{2_i}^2 + q_{3_i}^2} \le \sqrt{\frac{1-cos(\theta_{max})}{2}} \\
||W_i||_2 \le w_{max} \\
||U_i||_2 \le T_{max} \\
T_{min} \le \frac{\hat{U}_i^T}{||\hat{U}_i||_2} U_i \\
cos(\delta_{max})||U_i||_2 \le T_{z_i},\; i = 0,1,2,\ldots,p \\
\sigma^j = \frac{t_f^j - t_0^j}{2}, j = 0,1,2,\ldots,H-1 \\
t_f^j = t_0^{j+1}, j = 0,1,2,\ldots,H-2 \\
X_0^j = X_f^{j+1},\; j = 0,1,2,\ldots,H-2 \\
p = \sum_{j=0}^{H} N^j \\
\left|X - \hat{X}\right| \le \delta_x \; \left|U - \hat{U}\right| \le \delta_u
\end{cases} \tag{60}
$$

The problems P_{NLC} and P_{NLC}^{hp} can be addressed using convex optimization algorithms, such as the interior point method. However, these problems may remain unsolved if the trust region size is set too small and the accuracy of the initial guess is low. Conversely, a larger trust region may result in reduced solution accuracy. So, it is necessary to update the trust region size to obtain an optimal solution according to the quality of the initial guess.

4. Trust Region Update Strategy

4.1. Controlled Solution Process through Trust Region

Begin by setting initial guess $\hat{X}, \hat{U}$, and then solve P_{NLC}^{hp} to obtain solution X, U and objective value J. Then, update the trust region according to the system nonlinear cost Q and the system linear cost L. The system linear error is equal to the linearized dynamic error plus the linearized constraint error, while the system nonlinear error is equal to the nonlinear dynamic error plus the nonlinear constraint error. The calculation error ratio is denoted as follows:

$$
\rho = \frac{\left|\hat{Q} - Q\right|}{\left|\hat{Q} - L\right|} \tag{61}
$$

If C_1 and C_2 are nonlinear, they are linearized using a first-order Taylor series approximation. Q and L are calculated as follows:

$$\hat{Q} = ||D\hat{X} - \sigma f(\hat{X}, \hat{U})||_2 + ||C_1(\hat{X}, \hat{U})||_2 + ||C_2(\hat{X}, \hat{U})||_2 \tag{62}$$

$$Q = ||DX - \sigma f(X, U)||_2 + ||C_1(X, U)||_2 + ||C_2(X, U)||_2 \tag{63}$$

$$L = ||DX - \sigma(L(X, U) + R(\hat{X}, \hat{U}))||_2 + ||LC_1(\hat{X}, \hat{U}, X, U)||_2 + ||LC_2(\hat{X}, \hat{U}, X, U)||_2 \tag{64}$$

If C_1 C_2 are linear, they are calculated as follows:

$$\Delta Q = ||DX - \sigma f(X, U)||_2 \tag{65}$$

$$\Delta L = ||DX - \sigma(L(X, U) + R(\hat{X}, \hat{U}))||_2 \tag{66}$$

The trust region is updated based on the value of ρ. We set $\rho_0 = 0.1, \rho_1 = 0.25$, $\rho_2 = 0.9$. Here is how the trust region is updated depending on the value of ρ. If $\rho < \rho_0$, the current solutions are rejected. If $\rho_0 < \rho < \rho_1$, δ_x and δ_u will be scaled down. $\delta_x = \alpha\delta_x$, $\delta_u = \alpha\delta_u$, $0 < \alpha < 1$, and $\hat{X} = X, \hat{U} = U$. If $\rho_1 < \rho < \rho_2$, the trust region remains unchanged, and $\hat{X} = X, \hat{U} = U$. If $\rho_2 < \rho$, it indicates that the trust region should be enlarged. $\delta_x = \beta\delta_x, \delta_u = \beta\delta_u$, where $\beta > 1$, and $\hat{X} = X, \hat{U} = U$.

By incorporating the proposed trust region update method with the PCMs, we introduce the following two algorithms for solving the optimal control problem, specifically for the 6-Degree-of-Freedom powered landing.

We incorporate the method from Section 3.2 into Algorithm 1 to propose Algorithm 2 as follows.

Algorithm 1: Legendre Pseudospectral Convex Method with Trust Region Strategy (LPCM-TRS)

Input: a optimal control problem P, a number of collocation of points N, trust region size δ_x and δ_u, initial guessed solution $\hat{X}$ and $\hat{U}$, tolerance ϵ, Maximum number of iterations η, other parameters α, β, ρ_0, ρ_1, ρ_2
Output: X and U

Transforming problem P into problem P_c using Equations (26)–(35)
Transforming problem P_c into problem P_{NC} using Equations (36)–(59)
for each $i \in [1, \eta]$
    Getting solution X and U by solving P_{NC}
    Calculating ρ using equation(61)
  while True **do**
    if $\rho = 1$ or $||\delta_x||_2 \leq \epsilon$
      return X, U
    else
      if $\rho \leq \rho_0$ **then**
        $\delta_x = \delta_x / \alpha$
      else
        $\hat{X} = X, \hat{U} = U$
      if $\rho \leq \rho_1$ **then**
        $\delta_x = \delta_x / \alpha$
      else if $\rho \geq \rho_2$ **then**
        $\delta_x = \delta_x \cdot \beta$
      end if
      break
    end if
   end if
  end while
  update the problem P_{NC}
end for
return X, U

Algorithm 2: hp-Legendre Pseudospectral Convex Method with Trust Region Strategy (hp-LPCM-TRS)

Input: a optimal control convex problem P_{NC}, a total number of mesh H, a list of the number of points N_{hp}, a list of the interval time T_{hp}, other parameters δ_x, δ_u, $\hat{X}$, $\hat{U}$, ϵ, η, β, ρ_0, ρ_1, ρ_3
Output: X and U

for each $h \in [1, H]$ **do**
    $N = N_{hp}[h]$
    $t_0, t_f = T_{hp}[h]$
    To calculate pseudospectral differential matrix D using Equation (31)
    To create a equality constraints using Equation (52)
    To link next mesh using Equations (53) and (54)
end for
To transform P_{NC} into a multi-mesh problem P_{hpc}
for each $i \in [1, \eta]$ **do**
    Getting solution X and U by solving P_{hpc}
    Calculating ρ using Equation (61)
  while True **do**
      if $\rho = 1$ or $||\delta_x||_2 \leq \epsilon$ **then**
          return X,U
      else
          if $\rho \leq \rho_0$ **then**
              $\delta_x = \delta_x / \alpha$
          else
              $\hat{X} = X, \hat{U} = U$
              if $\rho < \rho_1$ **then**
                  $\delta_x = \delta_x / \alpha$
              else if $\rho \geq \rho_2$ **then**
                  $\delta_x = \delta_x \cdot \beta$
              end if
              break
          end if
        end if
    end while
    update the problem P_{hpc}
end for
return X,U

The convergence of the LPCM-TRS and hp-LPCM-TRS algorithms will be theoretically discussed in the next subsection.

4.2. Convergence Analysis

In reference [5], it has been demonstrated that the NLP P_N, obtained using the Legendre pseudospectral method, satisfies the KKT conditions. Building upon this foundation, we can analyze the KKT conditions for the relaxed SOCP P_C and the relaxed SOCP of mesh division P_{hpC}. The KKT conditions for problem P_N are as follows:

$$\nabla_x J + \sum_{k=0}^{N} \gamma_k \nabla_x \left(\sum_{i=0}^{N} D_{ki} X_i - f(X_k, U_k) \right) + \sum_{k=0}^{N} \eta_k \nabla_x C_1(X_k, U_k) + \sum_{k=0}^{N} \psi_k \nabla_x C_2(X_k, U_k) = 0 \tag{67}$$

$$\nabla_u J + \sum_{k=0}^{N} \gamma_k \nabla_u \left(\sum_{i=0}^{N} D_{ki} X_i - f(X_k, U_k) \right) + \sum_{k=0}^{N} \eta_k \nabla_u C_1(X_k, U_k) + \sum_{k=0}^{N} \psi_k \nabla_x C_2(X_k, U_k) = 0 \tag{68}$$

$$\sum_{k=0}^{N}\sum_{i=0}^{N} D_{ki}X_k = f(X_k, U_k) \tag{69}$$

$$\psi_k \geq 0, \psi_k \cdot C_1(X_k, U_k) = 0 \tag{70}$$

$$C_1(X_k, U_k) = 0 \tag{71}$$

$$C_2(X_k, U_k) \leq 0 \tag{72}$$

where γ_k, η_k, and ψ_k represent Lagrange multipliers and $k = 0, 1, .., N$. Assuming that C_1 and C_2 are linear, we will discuss the linearized dynamical equations. The KKT conditions for problem P_C are as follows:

$$\nabla_x J_c + \sum_{k=0}^{N} \gamma_k \nabla_x \left(\sum_{i=0}^{N} D_{ki}X_i - \left(L(X_k, U_k) + R\left(\hat{X}_k, \hat{U}_k\right) \right) \right) + \sum_{k=0}^{N} \eta_k \nabla_x C_1(X_k, U_k) + \sum_{k=0}^{N} \psi_k \nabla_x C_2(X_k, U_k) = 0 \tag{73}$$

$$\nabla_x J_c + \sum_{k=0}^{N} \gamma_k \nabla_u \left(\sum_{i=0}^{N} D_{ki}X_i - \left(L(X_k, U_k) + R\left(\hat{X}_k, \hat{U}_k\right) \right) \right) + \sum_{k=0}^{N} \eta_k \nabla_u C_1(X_k, U_k) + \sum_{k=0}^{N} \psi_k \nabla_u C_2(X_k, U_k) = 0 \tag{74}$$

$$\sum_{i=0}^{N} D_{ki}X_i - \left(L(X_k, U_k) + R\left(\hat{X}_k, \hat{U}_k\right) \right) - v_k = 0 \tag{75}$$

$$L(X_k, U_k) + R\left(\hat{X}_k, \hat{U}_k\right) = f\left(\hat{X}_k, \hat{U}_k\right) + \nabla_x f\left(\hat{X}_k, \hat{U}_k\right)\left(X_k - \hat{X}_k\right) + \nabla_u f\left(\hat{X}_k, \hat{U}_k\right)\left(U_k - \hat{U}_k\right) \tag{76}$$

$$\psi_k \geq 0, \psi_k \cdot C_1(X_k, U_k) = 0 \tag{77}$$

$$C_1(X_k, U_k) = 0 \tag{78}$$

$$C_2(X_k, U_k) \leq 0 \tag{79}$$

$$\left|X_k - \hat{X}_k\right| \leq \delta_x \tag{80}$$

$$\left|U_k - \hat{U}_k\right| \leq \delta_u \tag{81}$$

It can be seen that when $\delta_x = 0,\ \delta_u = 0,\ v_k = 0$, the KKT conditions of problem P_N are equivalent to the KKT conditions of problem P_C. The KKT conditions for problem P_{hpC} are as follows:

$$\begin{gathered}\nabla_x J_c + \sum_{s=0}^{H}\left(\sum_{k=0}^{N} \gamma_k^s \nabla_x \left(\sum_{i=0}^{N} D_{ki}^s X_i^s - \left(L\left(X_k^s, U_k^s\right) + R\left(\hat{X}_k^s, \hat{U}_k^s\right) \right) \right) \right) + \sum_{s=0}^{H}\sum_{k=0}^{N} \eta_k^s \nabla_x C_1\left(X_k^s, U_k^s\right)) \\ + \sum_{s=0}^{H}\sum_{k=0}^{N} \psi_k^s\ \nabla_x C_2\left(X_k^s, U_k^s\right) = 0\end{gathered} \tag{82}$$

$$\begin{gathered}\nabla_x J_c + \sum_{s=0}^{H}\left(\sum_{k=0}^{N} \eta_k^s \nabla_u \left(\sum_{i=0}^{N} D_{ki}^s X_i^s - \left(L\left(X_k^s, U_k^s\right) + R\left(\hat{X}_k^s, \hat{U}_k^s\right) \right) \right) \right) + \sum_{s=0}^{H}\sum_{k=0}^{N} \eta_k^s \nabla_u C_1\left(X_k^s, U_k^s\right)) \\ + \sum_{s=0}^{H}\sum_{k=0}^{N} \psi_k^s\ \nabla_u C_2\left(X_k^s, U_k^s\right) = 0\end{gathered} \tag{83}$$

$$\sum_{k=0}^{H}\sum_{i=0}^{N} D_{ki}^s X_i^s - \left(L\left(X_k^s, U_k^s\right) + R\left(\hat{X}_k^s, \hat{U}_k^s\right) \right) - v_k^s = 0 \tag{84}$$

$$L(X_k^s, U_k^s) + R\left(\hat{X}_k^s, \hat{U}_k^s\right) = f\left(\hat{X}_k^s, \hat{U}_k^s\right) + \nabla_x f\left(\hat{X}_k^s, \hat{U}_k^s\right)\left(X_k^s - \hat{X}_k^s\right) + \nabla_u f\left(\hat{X}_k^s, \hat{U}_k^s\right)\left(U_k^s - \hat{U}_k^s\right) \tag{85}$$

$$\psi_k^s \geq 0, \psi_k^s \cdot C_1(X_k^s, U_k^s) = 0 \tag{86}$$

$$C_1(X_k^s, U_k^s) = 0 \tag{87}$$

$$C_2(X_k^s, U_k^s) \leq 0 \tag{88}$$

$$\left|X_k^s - \hat{X}_k^s\right| \leq \delta_x \tag{89}$$

$$\left|U_k^s - \hat{U}_k^s\right| \leq \delta_u \tag{90}$$

where $s = 0, 1, \ldots, H$. Here, the superscript s indicates that it is in the s-th mesh interval, and it is evident that the KKT conditions of problem P_{hpc} consist of multiple mesh KKT conditions. Each mesh can be viewed as a scaled-down problem P_c.

To ensure that the KKT conditions of these problems are as equal as possible, the trust region updating based on the nonlinear error and linear error of the system is proposed. Then, its convergence is analyzed as follows:

When $\rho < 1$, Q is larger than the L error. It is necessary to scale down the trust region size to confine X and U to the vicinity of $\hat{X}$ and $\hat{U}$. In general, a solution is considered invalid if ρ is less than 0.25. When $\rho > 1$, it means that Q is smaller than L. In this case, the problem might be unsolvable, and there is a need to scale up the trust region size to search for a credible solution. When $\rho = 1$, it indicates that Q is equivalent to L. Only when $\hat{X} - X = 0$ and $\hat{U} - U = 0$, the solution of SOCP is the globally convergent solution of NLP.

The practical solution capability of the proposed method, although theoretically analyzed for optimality, will be verified through numerical experiments in the next section. P-iSc [7] is a pseudospectral convexification method based on the Radau pseudospectral approach, without mesh division. To facilitate a more meaningful comparison, we introduce mesh division to the original P-iSC algorithm, resulting in a new method called hp-P-iSc. We then proceed to compare it with LPCM-TRS and hp-LPCM-TRS.

5. Numerical Simulation

In this section, numerical experiments on 6-Degree-of-Freedom rocket landing were conducted to validate the effectiveness of the proposed method. The method's performance is compared with hp-P-iSC for solving the rocket-powered landing. The simulations were implemented using Python, and the simulation parameters and boundary conditions are presented in Table 1. The method's parameters are provided in Table 2.

The methods compared with hp-P-iSC include LPCM-TRS and hp-LPCM-TR. The total number of collocation points is 50, with each mesh containing 10 points, and a maximum of 20 iterations are allowed.

To facilitate a comprehensive comparison of the optimization results' accuracy, the discrete control and state quantities obtained from the optimization are utilized in the nonlinear dynamic equation. This yields discrete differential values, which are then integrated through time-stepping to produce continuous state quantity curves. To maintain consistency in the subsequent legends, the discrete numerical solutions in the figures are denoted by red triangles and green circles corresponding to hp-LPGM-TR and LPCM, respectively. The integrated continuous solutions are represented by red lines and green lines for hp-LPCM-TRS and LPCM-TRS, respectively. Additionally, the blue diamonds and lines represent the discrete numerical solutions and integrated continuous solutions of the hp-P-iSC method.

Table 1. Simulation parameters.

Parameter	Value	Unit
m_{wet}	3×10^4	kg
m_{dry}	2.2×10^4	kg
T_{max}	8×10^5	$\mathrm{kg} \cdot \mathrm{m/s^2}$
T_{min}	3.2×10^5	$\mathrm{kg} \cdot \mathrm{m/s^2}$
W_{max}	90	deg
r_b	[0,0,−14]	m
j_b	$[4 \times 10^6, 4 \times 10^6, 1 \times 10^5]$	$\mathrm{kg} \cdot \mathrm{m^2}$
R_0	[0,200,500]	m
V_0	[20,7,−78]	m/s
Q_0	[1,0,0,0]	-
W_0	[0,0,0]	deg
R_f	[0,0,0]	m
V_f	[0,0,−5]	m/s
Q_f	[1,0,0,0]	-
W_f	[0,0,0]	deg

Table 2. Algorithm parameters.

Parameter	Value	Unit
w_v	1×10^5	-
δ_x	5	-
δ_u	5	-
α	2.0	-
β	3.2	-
H	10	-
p	50	-

Observing the discrete points and continuous curves of the same-colored mass in Figure 1, it is evident that they generally align well. The difference between hp-P-iSC and hp-LPCM-TRS is minimal, with LPCM-TRS having the lowest performance index.

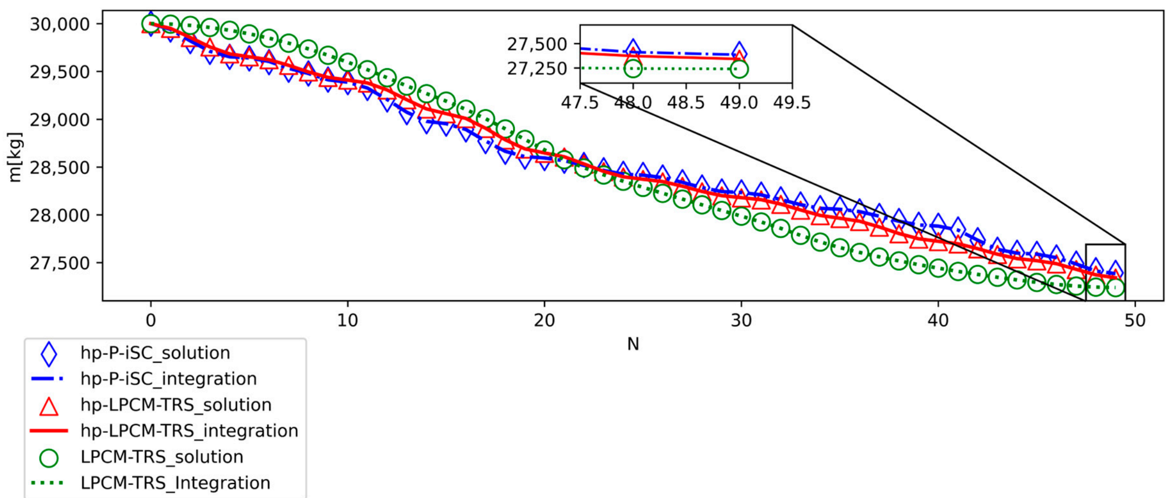


Figure 1. The mass curve of 6-Degree-of-Freedom powered landing.

Figure 2 illustrates the changes in the position coordinates of the rocket in three-dimensional space. In each of the three subgraphs, the discrete points and continuous curves exhibit a good fit. When examining their respective terminal positions, it is evident that LPCM-TRS gradually approaches the target point several time points in advance, displaying a smoother trajectory. On the other hand, hp-LPCM-TRS and hp-P-iSC reach the target point with steeper changes in their trajectories.

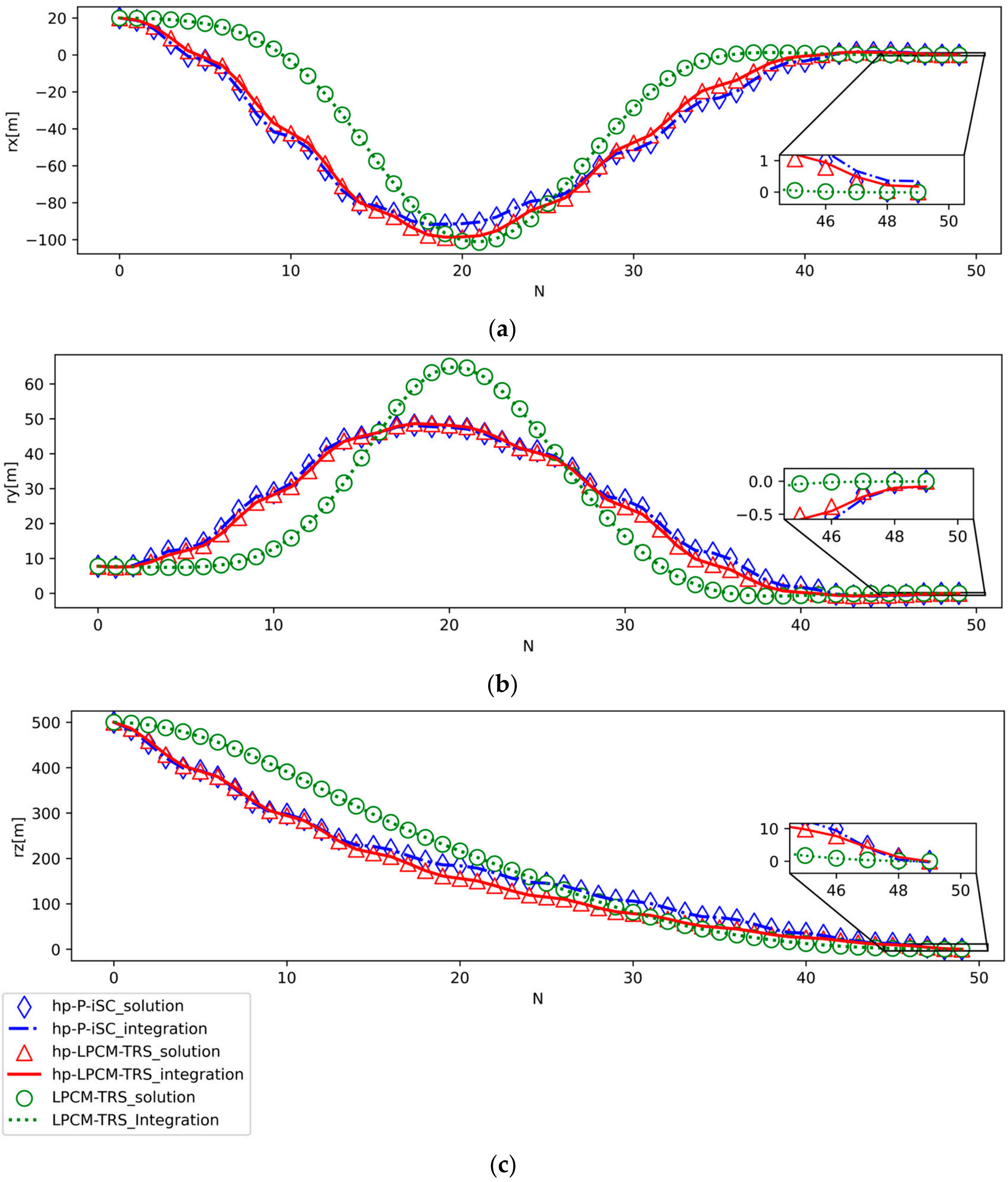
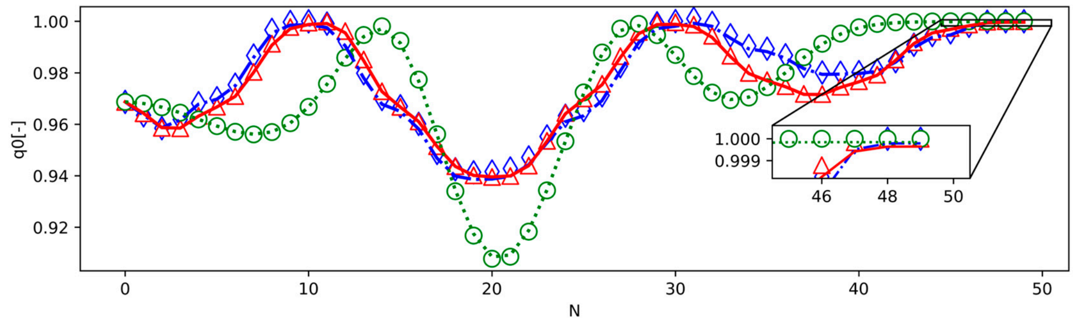
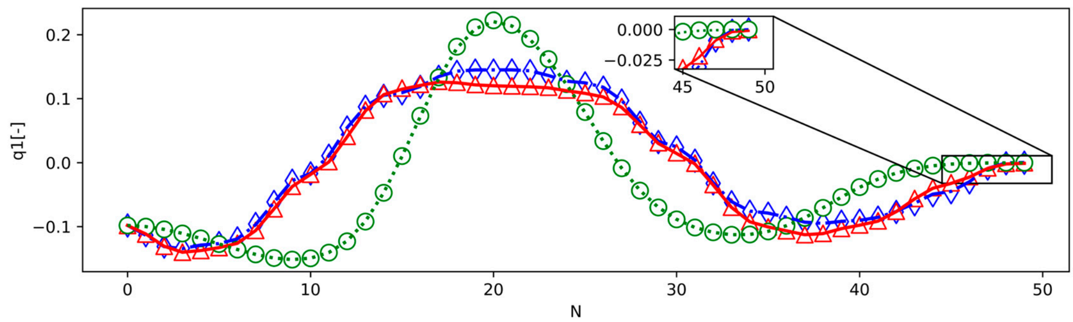


Figure 2. The changes in the position coordinates of the rocket in three dimensions are described as follows: (**a**) The position component r_x represents the changes along the x-axis. (**b**) The position component r_y represents the changes along the y-axis. (**c**) The position component r_z represents the changes along the z-axis.

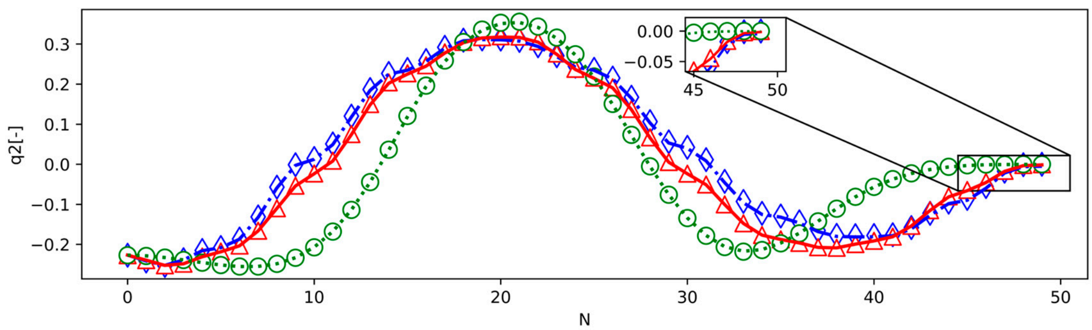
In Figure 3, the quaternion change curve for the rocket's control attitude change is depicted. The discrete points and continuous curves of q_0, q_1, and q_2 exhibit a good fit. However, in the case of q_3, there is a significant deviation in the middle part between the discrete points and the continuous curves for hp-P-iSC.



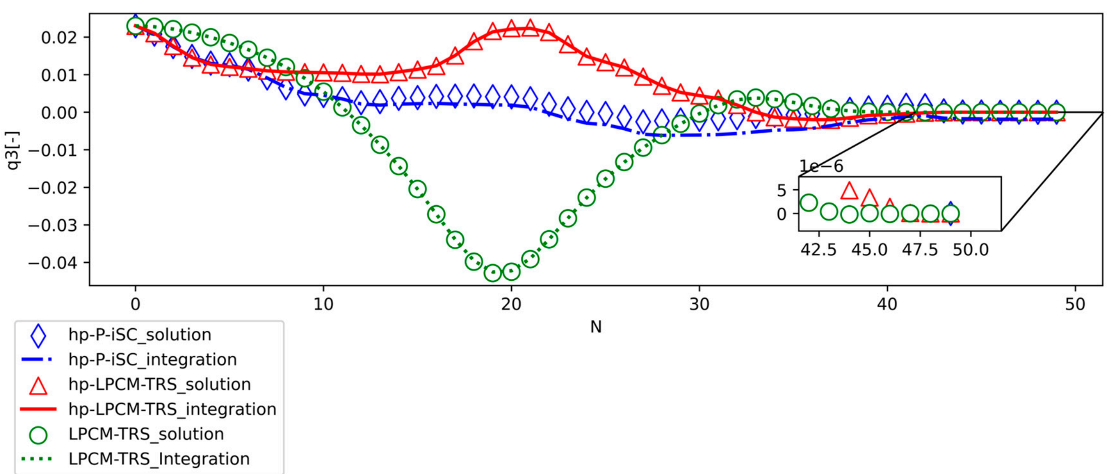
(**a**)



(**b**)



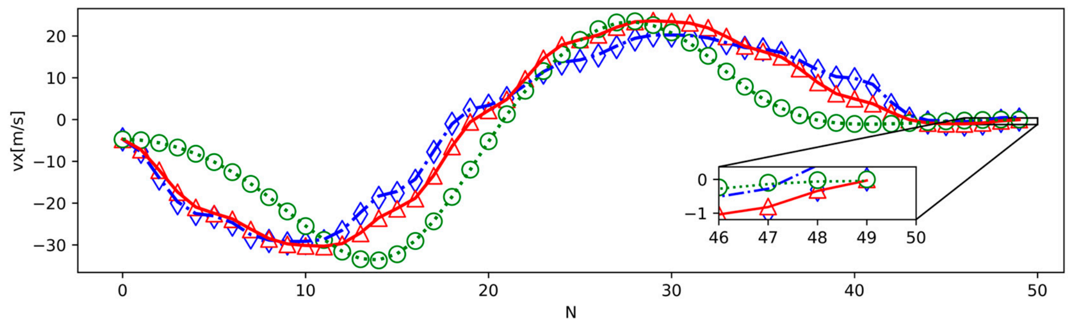
(**c**)



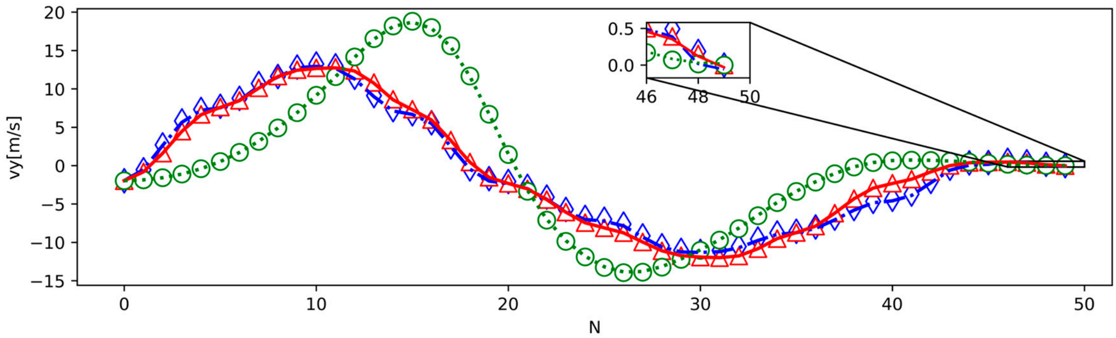
(**d**)

Figure 3. The quaternion change curve for rocket control attitude change is depicted in Figure 3, highlighting the changes in each component of the quaternion. The subgraphs are organized as follows: (**a**) Comparison of the solutions for the first component of quaternion q_0. (**b**) Comparison of the solutions for the second component of quaternion q_1. (**c**) Comparison of the solutions for the third component of quaternion q_2. (**d**) Comparison of the solutions for the last component of quaternion q_3.

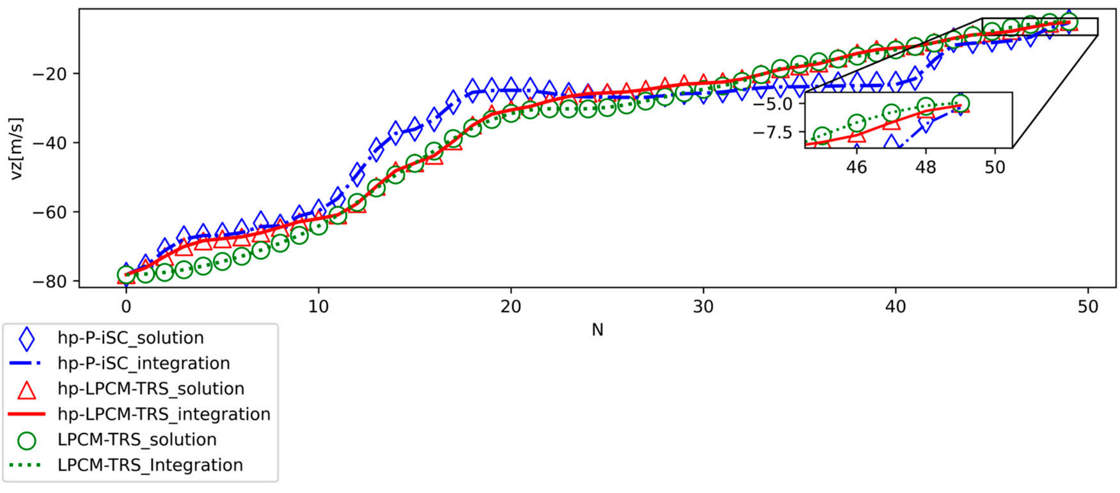
In Figure 4, we observe the fundamental alignment between discrete points and continuous curves in Figure 4a. However, it is noticeable that as we approach the end, there are deviations and discrepancies in the discrete points and continuous curves of hp-P-iSC. Interestingly, the positions of the discrete points in hp-P-iSC closely resemble those of hp-LPCM-TRS. In Figure 4b, the fitting between discrete points and continuous curves is excellent, and the trend remains consistent as we approach the end. Meanwhile, in Figure 4c, the changes in discrete points and continuous curves of hp-LPCM-TRS closely resemble those of LPCM-TRS, particularly near the end positions. However, hp-P-iSC exhibits a relatively steep change in trend as it approaches the end condition.



(**a**)



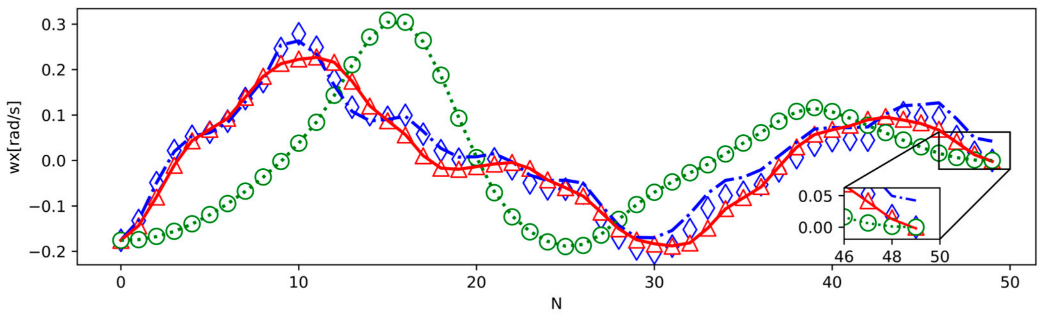
(**b**)



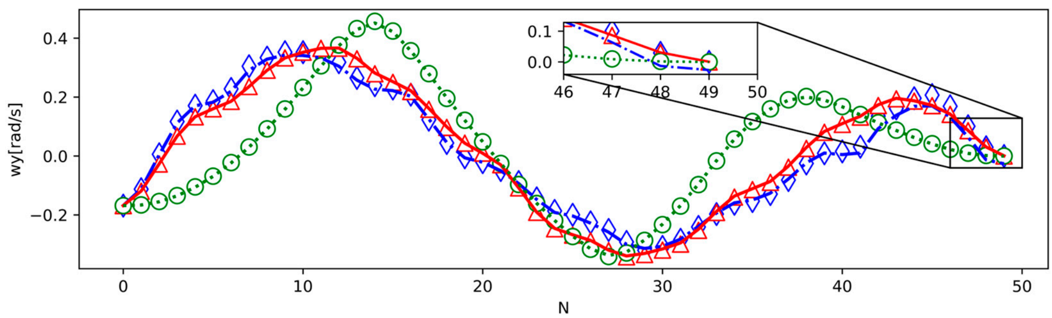
(**c**)

Figure 4. The speed of the rocket moving in three dimensions is illustrated. The three subgraphs represent (**a**) the speed of movement along the x-axis, (**b**) the speed of movement along the y-axis, and (**c**) the speed of movement along the z-axis.

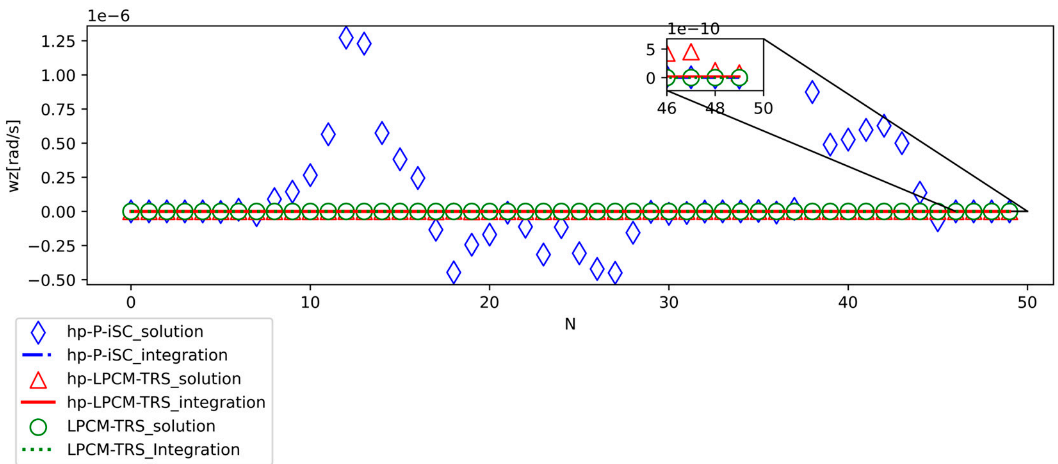
In Figure 5, there is a deviation in the fitting of discrete points and continuous curves for hp-P-iSC in Figure 5a, leading to a significant deviation of continuous curves at the end, despite the discrete points being consistent with hp-LPCM-TRS. Figure 5b shows a good fit between the discrete points and continuous curves. In Figure 5c, the value is small enough to be considered as zero.



(**a**)



(**b**)



(**c**)

Figure 5. The figures depict the variation in the rocket's angular velocity as it spins around its axis: (**a**) the angular velocity around the x-axis; (**b**) the angular velocity around the y-axis; (**c**) the angular velocity around the z-axis.

Table 3 presents various indicators, including the average error, objective function value, objective function error, relaxation value v_f, and CPU time. In this study, we have introduced two novel methods, namely LPCM-TRS and hp-LPCM-TRS, and conducted a comparative analysis with hp-P-iSC, which incorporates existing methodologies. As elucidated in the convergence analysis within Section 4.2, the proximity of the relaxation value to zero signifies a closer alignment between the solved problem and the original problem, ultimately resulting in a more dependable solution. In Table 3, it becomes evident that both LPCM-TRS and hp-LPCM-TRS exhibit relaxation values of 10^{-8}, while hp-P-iSC demonstrates a relaxation value of 10^{-5}. Despite hp-P-iSC boasting a shorter computing time, it regrettably fails to converge, as discerned from the algorithm's convergence analysis.

Table 3. The comparison of objective value, average error, and CPU time.

Method	Average Error	Objective Value	Objective Error	Relaxation Value v_f	CPU Time
LPCM-TRS	3.197×10^{-4}	9.080×10^{-1}	1.223×10^{-5}	1.041×10^{-8}	1.532×10^{2}
hp-LPCM-TRS	4.052×10^{-4}	9.114×10^{-1}	7.050×10^{-5}	2.093×10^{-8}	7.045×10^{1}
hp-P-iSC	2.771×10^{-3}	9.128×10^{-1}	3.007×10^{-4}	9.300×10^{-5}	5.692×10^{1}

Another pivotal metric to consider is the average error. It is noteworthy that both LPCM-TRS and hp-LPCM-TRS showcase remarkable precision in optimizing both spatial positioning and attitude control during rocket landing. Conversely, hp-P-iSC manifests a larger average error. This discrepancy can be attributed to the relatively significant attitude control errors in hp-P-iSC, while spatial position state accuracy remains consistently high, aligning with the findings presented in the original paper. Upon a comprehensive evaluation of the average error and objective function error indices, hp-LPCM-TRS and LPCM-TRS display comparable levels of accuracy,10^{-4} and 10^{-5}. Nevertheless, when evaluating CPU computing time, hp-LPCM-TRS remarkably expends only 45.9% of the time required by LPCM-TRS. Consequently, hp-LPCM-TRS emerges as the optimal choice, excelling in terms of accuracy, computational efficiency, and problem-solving optimality.

In summary, the optimal control problem of 6-Degree-of-Freedom rocket-powered landing was solved under the same iteration step and the same termination condition.

Based on the data from the Figures 1–5 and Table 3, it is apparent that hp-P-iSc has a relatively short calculation time. However, when examining the relaxation amount, it becomes evident that hp-P-iSC has a larger relaxation value compared to the other two algorithms. This larger relaxation value leads to relatively large average errors and objective function errors, ultimately resulting in a lack of solution accuracy. An analysis of the figures suggests that the poor calculation performance of hp-P-iSC is primarily due to the inadequate calculation of attitude control state variables.

The calculation results of hp-P-iSC in the spatial position state do not significantly differ from the other two algorithms. The average error and objective function error of LPCM-TRS and hp-LPCM-TRS are within the same order of magnitude, with very minimal actual differences between them. Moreover, the problem solved by the proposed algorithm closely approximates the original problem, given a relaxation value of 10^{-8}. Interestingly, the computational time of hp-LPCM-TRS is only 45.9% of the computational time required by LPCM-TRS.

6. Conclusions

The hp-LPCM-TRS method is introduced as an extension of the Legendre pseudospectral method for optimal control problems. In theory, it offers a general approach for solving linear optimal control problems, with the ability to address nonlinear optimal control problems by utilizing first-order Taylor series approximation for linearization. Specifically, the study focuses on solving the 6-Degree-of-Freedom rocket dynamic landing problem, creating and solving linear optimal control problems P_{NLC} and P_{NLC}^{hp} using LPCM-TRS, hp-LPCM-TRS, and hp-P-iSC.

The simulation results highlight that hp-LPCM-TRS emerges as an optimal choice for tackling the powered landing trajectory problem of a 6-Degree-of-Freedom rocket. It not only demands less computational time but also maintains a high level of accuracy in its solutions. The process of mesh division exhibits the potential to significantly enhance computational efficiency while upholding a commendable level of precision with an equivalent number of collocation points. The utilization of a trust region update strategy enables an iterative approach to closely approximate the original problem, thereby ensuring the robustness and fidelity of the solution. Within the framework of the Legendre pseudospectral method, PCM effectively preserves the coherence among multiple mesh partitions.

Author Contributions: Conceptualization, J.H.; Methodology, Y.Z.; Software, Y.Z.; Supervision, J.H.; Writing—Original Draft, Y.Z.; Writing—Review and Editing, J.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Natural Science Foundation of SiChuan under Grant 2022NSFSC0940, 2022NSFSC0446 and 2022NSFSC0894, and in part by Doctor Foundation of Southwest University of Science and Technology under Grant 20zx7137.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

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

Appendix A

$$A_m = [0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0]$$

$$B_m = \begin{bmatrix} -\frac{I_{sp}g_0 T_x}{\sqrt{T_Z^2+T_x^2+T_y^2}} & -\frac{I_{sp}g_0 T_y}{\sqrt{T_Z^2+T_x^2+T_y^2}} & -\frac{I_{sp}g_0 T_Z}{\sqrt{T_Z^2+T_x^2+T_y^2}} \end{bmatrix}$$

$$A_R = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B_R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$A_V = \begin{bmatrix} Av_{x_m} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2(T_z q_2 - T_y q_3)}{m} & \frac{2(T_z q_3 + T_y q_2)}{m} & \frac{2(T_z q_0 - 2T_x q_2 + T_y q_1)}{m} & \frac{2(T_z q_1 - 2T_z q_3 - T_y q_0)}{m} & 0 & 0 & 0 \\ Av_{y_m} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2(-T_z q_1 + T_z q_3)}{m} & \frac{2(-T_z q_0 + T_z q_2 - 2T_y q_1)}{m} & \frac{2(T_z q_3 + T_z q_1)}{m} & \frac{2(T_z q_2 + T_x q_0 - 2T_y q_3)}{m} & 0 & 0 & 0 \\ Av_{z_m} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2(-T_z q_2 + T_y q_1)}{m} & \frac{2(-2T_z q_1 + T_z q_3 + T_y q_0)}{m} & \frac{2(-2T_z q_2 - T_x q_0 + T_y q_3)}{m} & \frac{2(T_z q_1 + T_y q_2)}{m} & 0 & 0 & 0 \end{bmatrix}$$

$$Av_{x_m} = \frac{-2T_z(q_0q_2 + q_1q_3) + T_z\left(2q_2^2 + 2q_3^2 - 1\right) + 2T_y(q_0q_3 - q_1q_2)}{m^2}$$

$$Av_{y_m} = \frac{2T_z(q_0q_1 - q_2q_3) - 2T_z(q_0q_3 + q_1q_2) + T_y\left(2q_1^2 + 2q_3^2 - 1\right)}{m^2}$$

$$Av_{z_m} = \frac{T_z\left(2q_1^2 + 2q_2^2 - 1\right) + 2T_z(q_0q_2 - q_1q_3) - 2T_y(q_0q_1 + q_2q_3)}{m^2}$$

$$B_V = \begin{bmatrix} \frac{-2q_3^2 \pm 2q_3^2 + 1}{m} & \frac{2\left(-q_{0q_3} + q_{1q_2}\right)}{m} & \frac{2\left(q_{0q_3} + q_{1q_2}\right)}{m} \\ \frac{2\left(q_{0q_3} + q_{1q_2}\right)}{m} & \frac{-2q_3^2 - 2q_2^2 + 1}{m} & \frac{2\left(-q_{0q_3} + q_{2q_3}\right)}{m} \\ \frac{2\left(-q_{0q_2} + q_{1q_3}\right)}{m} & \frac{2\left(q_{0q_3} + q_2 q_3\right)}{m} & \frac{-2q_1^2 - 2q_2^2 + 1}{m} \end{bmatrix}$$

$$A_Q = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.5w_x & -0.5w_y & -0.5w_z & -0.5q_1 & -0.5q_2 & -0.5q_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5w_x & 0 & 0.5w_z & -0.5w_y & 0.5q_0 & -0.5q_3 & 0.5q_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5w_y & -0.5w_z & 0 & 0.5w_x & 0.5q_3 & 0.5q_0 & -0.5q_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5w_z & 0.5w_y & -0.5w_x & 0 & -0.5q_2 & 0.5q_1 & 0.5q_0 \end{bmatrix}$$

$$B_Q = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$A_W = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B_W = \begin{bmatrix} 0 & 52.5482732663404 & 0 \\ -52.5482732663404 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

References

1. Fahroo, F.; Ross, I.M. Direct Trajectory Optimization by a Chebyshev Pseudospectral Method. *J. Guid. Control Dyn.* **2002**, *25*, 160–166. [CrossRef]
2. Fahroo, F.; Ross, I.M. Advances in Pseudospectral Methods for Optimal Control. In Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit; Guidance, Navigation, and Control and Co-located Conferences, Honolulu, HI, USA, 18–21 August 2008; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2008.
3. Boyd, S.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004; ISBN 978-0-521-83378-3.
4. Kang, W.; Ross, I.M.; Gong, Q. Pseudospectral Optimal Control and Its Convergence Theorems. In *Analysis and Design of Nonlinear Control Systems: In Honor of Alberto Isidori*; Astolfi, A., Marconi, L., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 109–124. ISBN 978-3-540-74358-3.
5. Ross, I.M.; Fahroo, F. Legendre Pseudospectral Approximations of Optimal Control Problems. In *New Trends in Nonlinear Dynamics and Control and Their Applications*; Kang, W., Borges, C., Xiao, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2003; pp. 327–342.
6. Sagliano, M. Pseudospectral Convex Optimization for Powered Descent and Landing. *J. Guid. Control Dyn.* **2018**, *41*, 320–334. [CrossRef]
7. Wang, J.; Cui, N. A Pseudospectral-Convex Optimization Algorithm for Rocket Landing Guidance. In Proceedings of the 2018 AIAA Guidance, Navigation, and Control Conference, Kissimmee, FL, USA, 8–12 January 2018; AIAA SciTech Forum. American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2018.
8. Li, Y.; Chen, W.; Yang, L. Successive Chebyshev Pseudospectral Convex Optimization Method for Nonlinear Optimal Control Problems. *Int. J. Robust Nonlinear Control* **2022**, *32*, 326–343. [CrossRef]
9. Yu, C.-M.; Zhao, D.-J.; Yang, Y. Efficient Convex Optimization of Reentry Trajectory via the Chebyshev Pseudospectral Method. *Int. J. Aerosp. Eng.* **2019**, *2019*, 1414279. [CrossRef]
10. Zhang, Z.; Zhao, D.; Li, X.; Kong, C.; Su, M. Convex Optimization for Rendezvous and Proximity Operation via Birkhoff Pseudospectral Method. *Aerospace* **2022**, *9*, 505. [CrossRef]
11. Ross, I.M.; Fahroo, F. Pseudospectral Knotting Methods for Solving Nonsmooth Optimal Control Problems. *J. Guid. Control Dyn.* **2004**, *27*, 397–405. [CrossRef]
12. Sagliano, M. Generalized Hp Pseudospectral-Convex Programming for Powered Descent and Landing. *J. Guid. Control Dyn.* **2019**, *42*, 1562–1570. [CrossRef]
13. Lei, X.; Hongbo, Z.; Xiang, Z.; Guojian, T. Hp-Adaptive Pseudospectral Convex Optimization for Rocket Powered Landing Trajectory Planning. In Proceedings of the 2019 Chinese Automation Congress (CAC), Hangzhou, China, 22 November 2019; pp. 895–900.
14. Zhang, D.; Z–Hang, Y. PySCP: A Multiple-Phase Optimal Control Software Using Sequential Convex Programming. *Int. J. Aerosp. Eng.* **2022**, *2022*, 2969809. [CrossRef]
15. Szmuk, M.; Reynolds, T.; Acikmese, B.; Mesbahi, M.; Carson, J.M. Successive Convexification for 6-DoF Powered Descent Guidance with Compound State-Triggered Constraints. In Proceedings of the AIAA Scitech 2019 Forum, San Diego, CA, USA, 7–11 January 2019; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2019.