An improved adaptive hp mesh refinement method in solving optimal control problems

Changxin Luo | Jiong Li | Chijun Zhou | Humin Lei

Abstract
In order to improve the efficiency of the adaptive hp pseudospectral method in solving the optimal control problems, an improved adaptive hp pseudospectral method is proposed in this paper. This method mainly improves the mesh refinement process of the traditional hp pseudospectral method to reduce the number of mesh iterations, so as to achieve the purpose of shortening the solving time. The focus of this method is twofold. Firstly, by judging the position of the sudden change points of control curve, the mesh is refined in advance around them. Secondly, the non-uniform segmentation iteration is carried out by using the curvature of the system state curve as the criterion. These two points together improve the solving efficiency of the proposed method. Through the convergence analysis and simulation verification, it is shown that the proposed method can effectively solve the optimal control problem, and it is also proved that the proposed method has higher solving efficiency than the traditional adaptive hp pseudospectral method.

KEYWORDS
adaptive hp method, mesh refinement, nonlinear programming problem, optimal control, pseudospectral method

1 INTRODUCTION

Recently, the direct collocation method represented by the pseudospectral method has become an important method to solve complex optimal control problems. The pseudospectral method mainly uses Lagrangian interpolation polynomials based on Legendre-Gauss (LG) points, Legendre-Gauss-Radau (LGR) points, or Legendre-Gauss-Lobatto (LGL) points to transform the optimal control problem into nonlinear programming problem (NLP), and then use the NLP solver to solve it.

In solving the optimal control problem, many scholars have actively studied the related theory and application of pseudospectral method. For three different types of collocation points (LG, LGR and LGL), Garg et al. study the theoretical foundation and algorithm framework of pseudospectral method, which lays an important foundation for the research of this algorithm. In reference 4, a unified framework of pseudospectral method for three collocation strategies is proposed, and the characteristics of the algorithm under each strategy are deeply analyzed. For the trajectory optimization problem in finite-horizon and infinite-horizon, the accuracy of solution of pseudospectral method based on LGR collocation points is studied in reference 5. For the infinite-horizon optimization control problem, Garg et al. study the influence of several different time domain transformation methods on the solution results of pseudospectral
method based on the two collocation points of LG and LGR. In order to improve the solving efficiency of NLP, Patterson et al.\textsuperscript{7} propose a method to calculate the first and second derivatives of the NLP function. For the optimal control problem with terminal constraints, Fahroo et al.\textsuperscript{10} reveal the superiority of the pseudospectral method based on Legendre polynomial in dealing with this type of problem. Using the covector mapping principle, Gong et al.\textsuperscript{11,12} prove that the discrete solution obtained by the pseudospectral method can converge to a continuous optimized solution.

However, if only a fixed number of collocation points and a single mesh interval are used, the pseudospectral method will be difficult to obtain a satisfactory solution for complex optimal control problems. In order to solve this problem, many scholars have put forward two ideas. One is the p method,\textsuperscript{13} which improves the solving precision by increasing the collocation points. The other is the h method,\textsuperscript{14–16} which increases the number of mesh intervals by continuously dividing the time domain to ensure the solving precision. According to these two ideas, many scholars have made meaningful exploration. If there is a smooth analytical solution to the optimal control problem, the p method can get good results. Hussaini et al.\textsuperscript{17–19} has shown that in this case, by increasing the collocation points, a good approximation of the analytical solution and the exponential convergence rate can be achieved. However, if the solution of the optimal control problem is non-smooth, such as there are discontinuities or sudden changes, in this case, even if the number of collocation points is increased to increase the order of the interpolation polynomial, it is difficult to obtain a satisfactory solution. In response to this difficulty, h method is proposed, where original time domain is divided and analytic solution is approximated by establishing different mesh intervals. Relevant research has proved that this method can effectively make up for the shortcomings of p method.\textsuperscript{14–16}

Considering that the h method and the p method have their own advantages for different types of questions, some scholars have appropriately combined the two to further improve the solving precision and efficiency.\textsuperscript{20–23} Patterson et al.\textsuperscript{20} propose the ph method, which is actually a mesh refinement strategy using p method first and then h method. In other words, when using the ph method for mesh refinement, the order of the interpolation polynomial will first be increased to improve the solving precision. Then, if the order reaches the specified maximum limit, the mesh will be segmented instead until a satisfactory solution is obtained. Unlike ph method, a mesh refinement method called hp method is proposed in references 21–23, which adaptively decide whether to use h method or p method to refine the mesh through certain criteria. Darby et al.\textsuperscript{21} take the consistency of approximation error as the criterion, that is, if the approximation error of each collocation point is relatively consistent, the p method is used to refine mesh, otherwise, the h method is used. Different from this, Liu et al.\textsuperscript{22,23} take the smoothness of the solution as the criterion, that is, if the obtained solution is smooth, the h method is used for refinement, otherwise, the p method is used. Due to the efficient solving ability and the flexibility of the h algorithm, its related theories and applications have been widely studied by many scholars.\textsuperscript{24–28} Li et al.\textsuperscript{29,30} apply the adaptive hp method to solve trajectory optimization problem of missile, which obtain satisfactory solving precision. Considering obstacles avoidance and overload minimization, Xia et al.\textsuperscript{31} propose an hp-pseudospectral sequential convex programming method to achieve trajectory optimization of unmanned aerial vehicle. Based on adaptive hp method, Xu et al.\textsuperscript{32} complete the optimal motion planning of the space manipulator. Liu et al.\textsuperscript{33} use hp-adaptive method realize path-tracking of vehicle. Yazdani et al.\textsuperscript{34} analyze the feasibility of hp-adaptive Radau pseudospectral method. By improving the mesh refinement process, some effective variants of the hp method are proposed in references 35–39. In addition, several numerical methods\textsuperscript{40–43} are proposed, which also provide important ideas for solving various types of optimal control problem.

Although the aforementioned works\textsuperscript{21–39} on the adaptive hp method has made great progress, there still are several issues that need further consideration.

1. When there are sudden change points in the optimal control curve (such as bang-bang control\textsuperscript{44,45} switch control\textsuperscript{46}), in order to obtain a satisfactory solution, some existing adaptive hp methods often configure a large number of collocation points near the sudden change point, or segments the mesh frequently, which largely increases the dimensions of the NLP, therefore, affects the computational efficiency.

2. Most of the existing hp methods divide the mesh evenly according to the criterion, which is inefficient and may lead to frequent segmentation in the process of mesh refinement. This process will inevitably increase the number of iterations and prolong the computing time.

3. When hp method is used to solve optimal control problem with Bolza form, most existing works do not give the strict proof of convergence of this algorithm. Although the convergence of hp method is analyzed in reference 52, it is only discussed for solving unconstrained optimal control problems.
In view of the aforementioned consideration, this paper proposes an improved adaptive hp mesh refinement method, which mainly has three contributions.

1. By accurately locating the sudden change points of the control curve, the mesh points are added in the vicinity of these points. With the targeted mesh refinement in advance, the number of iterations of the grid is reduced and the efficiency of the algorithm is improved.

2. In order to further improve the efficiency of mesh refinement, a mesh segmentation method based on the cumulative sum of state curvature values is designed. This non-uniform segmentation method (unlike the uniform mesh segmentation method in references 21–24) can ensure that the locations that need more refinement can be quickly allocated to more mesh points.

3. For the Bolza optimization control problem with equality and inequality constraints, we extend the results in reference 52 and further prove the convergence of the improved adaptive hp algorithm proposed in this paper.

The remaining part of the paper is organized as follows. The Section 2 is the problem formulation, which describes the optimal control problem to be solved in this paper. In the Section 3, based on LGR collocation method, the original optimal control problem is transformed into a nonlinear programming problem. The Section 4 is the description of the improved adaptive hp mesh refinement method, and the core algorithm of this paper is introduced in detail. In Section 5, we prove the convergence of the proposed algorithm. The simulation example in the Section 6 verifies the effectiveness and superiority of the method in this paper. In the Section 7, the work of this paper is summarized.

## 2 PROBLEM FORMULATION

Without loss of generality, we define an optimal control problem with Bolza form as in (1), that is, to find the state-control function \((x, u) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_u}\) such that

\[
\begin{align*}
(B) \quad & \min \quad J = E(x_0, x_f, t_0, t_f) + \int_{t_0}^{t_f} F(x(t), u(t)) \, dt \\
& \text{s.t.} \quad \dot{x}(t) = f(x(t), u(t)) \\
& \quad e(x_0, x_f, t_0, t_f) = 0 \\
& \quad h(x(t), u(t)) \leq 0
\end{align*}
\]

Where, \(x \in \mathbb{R}^{n_x}\) is the state variable, \(u \in \mathbb{R}^{n_u}\) is the control variable, and \(x\) and \(u\) are defined as row vectors. \(t_0\) and \(t_f\) represent initial time and terminal time, respectively, and \(x_0 = x(t_0), x_f = x(t_f)\). \(E(\cdot)\) and \(F(\cdot)\) are scalar functions, which represent the terminal type and integral type performance indicators, respectively. \(f(\cdot) \in \mathbb{R}^{n_x}\) represents dynamic constraints, \(e(\cdot) \in \mathbb{R}^{n_e}\) represents boundary conditions, \(h(\cdot) \in \mathbb{R}^{n_h}\) represents inequality path constraints. Define \(f, e\) and \(h\) as row vectors. All these nonlinear functions \((E, F, f, e, h)\) are continuously differentiable in their domains, and their gradients are Lipschitz continuous.

According to Pontryagin minimum principle, we can reduce the first-order necessary condition of problem B to the following boundary-value problem.

\[
\begin{align*}
(B^*) \quad & \text{find} \quad [x(\cdot), u(\cdot), \dot{x}(\cdot), \mu(\cdot), \nu(\cdot)] \\
& \text{such that} \quad \dot{x}(t) = f(x(t), u(t)) \\
& \quad e(x_0, x_f, t_0, t_f) = 0 \\
& \quad h(x(t), u(t)) \leq 0 \\
& \quad \dot{\mu}(t) = -\partial \tilde{H}(t)/\partial x \\
& \quad \partial \tilde{H}/\partial u = 0 \\
& \quad \{\lambda(t_0), \lambda(t_f)\} = \{\partial \tilde{E}/\partial x_0, \partial \tilde{E}/\partial x_f\} \\
& \quad \{H(t_0), H(t_f)\} = \{\partial \tilde{E}/\partial t_0, -\partial \tilde{E}/\partial t_f\} \\
& \quad \mu(t) \geq 0, \mu(t) \cdot h^T(x(t), u(t)) = 0
\end{align*}
\]
Where $\bar{H}(t)$ represents the Lagrangian of the Hamiltonian.

$$\bar{H}(\lambda, \mu, x, u) := H(\lambda, x, u) + \mu \cdot h^T(x, u)$$  \hspace{1cm} (3)

$H$ is the Hamiltonian function.

$$H(\lambda, x, u) := F(x, u) + \lambda : f^T(x, u)$$  \hspace{1cm} (4)

$\bar{E}$ represents the Lagrangian form of the endpoint constraint.

$$\bar{E}(v, x_0, x_f, t_0, t_f) := E(x_0, x_f, t_0, t_f) + ve^T(x_0, x_f, t_0, t_f)$$  \hspace{1cm} (5)

In addition, the Lagrangian multipliers $\mu, \lambda$, and $v$ in Equations (3)–(5) are row vectors of $n_h$ dimension, $n_f$ dimension and $n_e$ dimension, respectively.

In order to solve the problem B, here we adopt the idea of mesh division in the hp pseudospectral method, select $K + 1$ mesh points $[T_0, T_1, \ldots, T_K]$ on the entire time interval $[t_0, t_f]$, and divide them into $K$ mesh intervals $S_k = [T_{k-1}, T_k], \quad k = 1, 2, \ldots, K$. Where $T_0 = t_0$, $T_K = t_f$, $\bigcup_{k=1}^K S_k = [t_0, t_f]$. Let $x^{(k)}(t)$ and $u^{(k)}(t)$ denote the state and the control on the mesh interval $S_k$, respectively. Therefore, problem B Equation (1) can be equivalent to

$$\begin{aligned}
\min & \quad J = E(x^{(1)}(T_0), x^{(K)}(T_K)) + \sum_{k=1}^K \int_{t_{k-1}}^{T_k} F(x^{(k)}(t), u^{(k)}(t)) \, dt \\
\text{s.t.} & \quad \frac{dx^{(k)}(t)}{dt} = f(x^{(k)}(t), u^{(k)}(t)) \quad k = 1, 2, \ldots, K \\
& \quad e(x^{(1)}(T_0), x^{(K)}(T_K)) = 0 \\
& \quad h(x^{(k)}(t), u^{(k)}(t)) \leq 0 \quad k = 1, 2, \ldots, K
\end{aligned}$$  \hspace{1cm} (6)

Since the state must be continuous in time interval $[t_0, t_f]$, the following conditions should be satisfied at each internal mesh point.

$$x^{(k)}(T^+_k) = x^{(k+1)}(T^-_k), \quad k = 1, 2, \ldots, K - 1$$  \hspace{1cm} (7)

### 3. Legendre-Gauss-Radau Collocation

In order to transform problem B into nonlinear programming problem (NLP), here we use the LGR collocation method\textsuperscript{4–6} to discretize Equation (6). First of all, we need to introduce time-domain variation to transform time $t$ of each grid interval $S_k = [T_{k-1}, T_k], \quad k = 1, 2, \ldots, K$ to interval $[-1, 1]$. Introduce a new time variable $\tau$ and make the transformation as shown below.

$$\tau = \frac{2t}{T_k - T_{k-1}} - \frac{T_k + T_{k-1}}{T_k - T_{k-1}} \quad k = 1, 2, \ldots, K$$  \hspace{1cm} (8)

Let $h_k$ denote the length of each mesh interval $S_k$, that is, $h_k = T_k - T_{k-1}$. Combined with Equations (6) ~ (8), problem B can be rewritten as

$$\begin{aligned}
\min & \quad J = E(x^{(1)}(-1), x^{(K)}(+1)) + \sum_{k=1}^K h_k \int_{-1}^{+1} F(x^{(k)}(\tau), u^{(k)}(\tau)) \, d\tau \\
\text{s.t.} & \quad \frac{dx^{(k)}(\tau)}{d\tau} = \frac{h_k}{2} f(x^{(k)}(\tau), u^{(k)}(\tau)) \quad k = 1, 2, \ldots, K \\
& \quad e(x^{(1)}(-1), x^{(K)}(+1)) = 0 \\
& \quad h(x^{(k)}(\tau), u^{(k)}(\tau)) \leq 0 \quad k = 1, 2, \ldots, K \\
& \quad x^{(k)}(+1) = x^{(k+1)}(-1), \quad k = 1, 2, \ldots, K - 1
\end{aligned}$$  \hspace{1cm} (9)
Then, select \( N_k \) LGR collocation points \( r_1^{(k)}, \ldots, r_{N_k}^{(k)} \in [-1, 1] \) on each mesh interval \( S_k(k = 1, \ldots, K) \), where \( r_1^{(k)}, \ldots, r_{N_k}^{(k)} \) \( (r_1^{(k)} = -1) \) is the root of polynomial \( P_{N_k-1} + P_{N_k} \) (\( P_{N_k} \) is Legendre polynomial of order \( N_k \)). In order to discretize Equation (9), we use LGR collocation points \( r_1^{(k)}, \ldots, r_{N_k}^{(k)} \) and noncollocated point \( r_{N_k+1}^{(k)} = +1 \) as sampling points on each mesh interval \( S_k \) to achieve Lagrange interpolation on \( x^{(k)}(r) \). The interpolation polynomial based on these \( N_k + 1 \) points is expressed as follows

\[
\ell_j^{(k)}(r) = \prod_{i=1 \atop i \neq j}^{N_k+1} \frac{r - r_i^{(k)}}{r_j^{(k)} - r_i^{(k)}}, \quad j = 1, 2, \ldots, N_k + 1
\]

Thus, the continuous state \( x^{(k)}(r) \) on mesh interval \( S_k \) can be approximated as

\[
x^{(k)}(r) \approx X^{(k)}(r) = \sum_{j=1}^{N_k+1} X_j^{(k)} \ell_j^{(k)}(r)
\]

(11)

Here, \( X_j^{(k)}, j = 1, 2, \ldots, N_k + 1 \) is an approximation of \( x^{(k)}(r_j^{(k)}) \). Taking the derivative of Equation (11) at the collocation points \( r_1^{(k)}, \ldots, r_{N_k}^{(k)} \), we have

\[
\dot{X}^{(k)}(r_j^{(k)}) = \sum_{j=1}^{N_k+1} X_j^{(k)} \ell_j^{(k)}(r_j^{(k)}) = \sum_{j=1}^{N_k+1} X_j^{(k)} D_{ij}^{(k)}, \quad i = 1, 2, \ldots, N_k
\]

(12)

Where, \( D_{ij}^{(k)} = \ell_i^{(k)}(r_j^{(k)}) \). Let \( U_{ij}^{(k)}, j = 1, \ldots, N_k \) denote the approximation of the control \( u^{(k)}(r_j^{(k)}) \). Then, by interpolating the function \( F(x^{(k)}(r), u^{(k)}(r)) \), its integral can be approximated as

\[
\int_{-1}^{+1} F(x^{(k)}(r), u^{(k)}(r)) \, dr \approx \sum_{j=1}^{N_k} F(X_j^{(k)}, U_j^{(k)}) \int_{-1}^{+1} \ell_j^{(k)}(r) \, dr = \sum_{j=1}^{N_k} F(X_j^{(k)}, U_j^{(k)}) \omega_j^{(k)}
\]

(13)

Where quadrature weight \( \omega_j^{(k)} \) is

\[
\omega_j^{(k)} = \int_{-1}^{+1} \ell_j^{(k)}(r) \, dr
\]

(14)

Combining Equations (10)–(14), we can transform the optimal control problem (Equation (9)) into an NLP, which is given as problem \( B^N \)

\[
(B^N) \quad \begin{align*}
\min \quad & J = E \left( X_1^{(1)}, X_{N_k+1}^{(K)} \right) + \sum_{k=1}^{K} \sum_{j=1}^{N_k} h_{ij} \omega_j^{(k)} F \left( X_j^{(k)}, U_j^{(k)} \right) \\
\text{s.t.} \quad & \sum_{j=1}^{N_k+1} X_j^{(k)} D_{ij}^{(k)} = \frac{h_k}{2} f \left( X_j^{(k)}, U_j^{(k)} \right) \quad i = 1, 2, \ldots, N_k \quad k = 1, 2, \ldots, K \\
& e \left( X_1^{(1)}, X_{N_k+1}^{(K)} \right) = 0 \\
& h \left( X_1^{(k)}, U_1^{(k)} \right) \leq 0 \quad i = 1, 2, \ldots, N_k \quad k = 1, 2, \ldots, K \\
& X_{N_k+1}^{(k)} = X_1^{(k+1)}, \quad k = 1, 2, \ldots, K - 1
\end{align*}
\]

(15)

For problem \( B^N \), we can use the NLP solver SNOPT\(^9\) to solve it. However, the optimal solution solved directly may not be able to meet the accuracy requirements of the original problem \( B \). Therefore, we need to continuously adjust the position of mesh points \( [T_0, T_1, \ldots, T_K] \) (h method) and the number of LGR collocation points in each mesh interval (P method),
and then iteratively solve problem $B^N$ until the optimal solution meets the accuracy of the original problem $B$. In this paper, we propose an improved adaptive hp mesh refinement method.

## 4 | IMPROVED ADAPTIVE HP MESH REFINEMENT METHOD

For some optimal control problems, the sudden changes of the control and the excessive curvature of the state curve will bring difficulties to the mesh refinement process of the adaptive hp method. However, in most of the existing studies, the traditional h method is used to segment the mesh evenly, which leads to frequent mesh refinement, increasing the number of iterations and reducing the solving efficiency. In response to these problems, this paper has improved the traditional adaptive hp mesh refinement method and formed a new algorithm with two core points. One is to find out the interval containing control sudden changes by evaluating the derivative $\hat{U}^{(k)}_l$ at different allocation points, and then refine the interval emphatically. Second, different from the uniform segmentation method adopted by the traditional adaptive hp method, this paper determines the location of the mesh points according to the curvature of the state curve, that is, where the curvature is large, the mesh points are dense, otherwise they are sparse.

This section mainly introduces the improved adaptive hp mesh refinement method proposed in this paper. Section 4.1 is the evaluation of the solution error. In Section 4.2, we give a detailed description of the core algorithm of mesh refinement proposed in this paper.

### 4.1 | Evaluation of solution error

After solving the solution of problem $B^N$ using the NLP solver, it is also necessary to judge whether the solution can achieve satisfactory accuracy, that is, whether the error between $X^{(k)}(\tau)$ and $x^{(k)}(\tau)$ in Equation (11) can be within a tolerance range $\varepsilon$. Since we cannot know the analytical solution $x^{(k)}(\tau)$, we take the following method to evaluate the solution error in each mesh interval.

First, add an LGR collocation point in each grid interval. For the mesh interval $S_k (k = 1, 2, \ldots, K)$, denote the new $\Gamma_k$ collocation points as $\{\hat{\tau}_1^{(k)}, \ldots, \hat{\tau}_N^{(k)}\}$, where $\Gamma_k = N_k + 1$, $\hat{\tau}_1^{(k)} = -1$. At the same time, select noncollocated point $\hat{\tau}_{\Gamma_k+1}^{(k)} = +1$.

Then, based on the numerical solution $\left(X_j^{(k)}, U_j^{(k)}\right)$ and the corresponding sampling points $\left(\tau_1^{(k)}, \ldots, \tau_{N_k+1}^{(k)}\right)$, the state $X^{(k)}\left(\hat{\tau}_i^{(k)}\right)$ at $\left(\hat{\tau}_1^{(k)}, \ldots, \hat{\tau}_i^{(k)}, \ldots, \hat{\tau}_{\Gamma_k+1}^{(k)}\right)$ and the control $U^{(k)}\left(\hat{\tau}_i^{(k)}\right)$ at $\left(\hat{\tau}_1^{(k)}, \ldots, \hat{\tau}_i^{(k)}, \ldots, \hat{\tau}_{\Gamma_k+1}^{(k)}\right)$ are obtained by Lagrange interpolation. Therefore, $X^{(k)}\left(\hat{\tau}_i^{(k)}\right)$ can be approximately expressed as

$$X^{(k)}\left(\hat{\tau}_i^{(k)}\right) = \sum_{j=1}^{N_k+1} X_j^{(k)} \prod_{\substack{i \neq j \quad j \neq \Gamma_k+1 \quad \Gamma_k+1 \quad i = 1, \ldots, \Gamma_k+1}} (\hat{\tau}_i^{(k)}) \\
\hat{\tau}_j^{(k)}(\tau) = \prod_{i=1}^{N_k+1} \frac{\tau-\tau_i^{(k)}}{\hat{\tau}_j^{(k)}-\tau_i^{(k)}}, \quad j = 1, 2, \ldots, N_k + 1$$

Similarly, $U^{(k)}\left(\hat{\tau}_i^{(k)}\right)$ can be approximately expressed as

$$U^{(k)}\left(\hat{\tau}_i^{(k)}\right) = \sum_{j=1}^{N_k} U_j^{(k)} \prod_{\substack{i \neq j \quad \Gamma_k+1 \quad i = 1, \ldots, \Gamma_k}} (\hat{\tau}_i^{(k)}) \\
\hat{\tau}_j^{(k)}(\tau) = \prod_{i=1}^{N_k} \frac{\tau-\tau_i^{(k)}}{\hat{\tau}_j^{(k)}-\tau_i^{(k)}}, \quad j = 1, 2, \ldots, N_k$$

Finally, combining the $X^{(k)}\left(\hat{\tau}_i^{(k)}\right)$ and $U^{(k)}\left(\hat{\tau}_i^{(k)}\right)$ obtained above, integrate again to obtain the state $\hat{X}^{(k)}\left(\hat{\tau}_i^{(k)}\right)$ at $\left(\hat{\tau}_1^{(k)}, \ldots, \hat{\tau}_i^{(k)}, \ldots, \hat{\tau}_{\Gamma_k+1}^{(k)}\right)$ using the integral form of the discrete state equation.
For the new LGR collocation points \((\hat{\xi}_1^{(k)}, \ldots, \hat{\xi}_{n_x}^{(k)})\), construct a \(\Gamma_k \times (\Gamma_k + 1)\) differentiation matrix \(\hat{D}^{(k)}\), and the element at position \((i, j)\) is \(\hat{D}_{ij}^{(k)} = \hat{d}_j^{(k)}(\hat{\xi}_i^{(k)})/dr\), where

\[
\hat{D}_{ij}^{(k)}(\tau) = \prod_{l = 1}^{\Gamma_k + 1} \frac{\tau - \hat{\xi}_i^{(k)}}{\hat{\xi}_j^{(k)} - \hat{\xi}_i^{(k)}}, \quad j = 1, 2, \ldots, \Gamma_k + 1
\]

Similar to the derivation of the discrete state equation in Equation (15), the discrete state equation at collocation points \((\hat{\xi}_1^{(k)}, \ldots, \hat{\xi}_{n_x}^{(k)})\) can be expressed as follows

\[
\sum_{j=1}^{\Gamma_k+1} \hat{D}_{ij}^{(k)} X^{(k)}(\hat{\xi}_j^{(k)}) = \frac{h_k}{2} f\left(X^{(k)}(\hat{\xi}_i^{(k)}), U^{(k)}(\hat{\xi}_i^{(k)})\right), \quad i = 1, 2, \ldots, \Gamma_k, \quad k = 1, 2, \ldots, K
\]

At the same time, an \(\Gamma_k \times \Gamma_k\) integration matrix \(\hat{I}^{(k)} = [\hat{D}_1^{(k)}, \cdots, \hat{D}_{\Gamma_k+1}^{(k)}]^{-1}\) is constructed, where \(\hat{D}_i^{(k)}\) \((i = 2, \ldots, \Gamma_k + 1)\) is the \(i\) th column element of the differentiation matrix \(\hat{D}^{(k)}\). Therefore, combined with Equation (19), we can obtain the integral form of the discrete state equation as follows

\[
\hat{X}^{(k)}\left(\hat{\xi}_{i+1}^{(k)}\right) = X^{(k)}\left(\hat{\xi}_i^{(k)}\right) + \frac{h_k}{2} \sum_{j=1}^{\Gamma_k} \hat{D}_{ij}^{(k)} f\left(X^{(k)}\left(\hat{\xi}_j^{(k)}\right), U^{(k)}\left(\hat{\xi}_j^{(k)}\right)\right), \quad i = 1, \ldots, \Gamma_k
\]

Where \(\hat{\xi}_j^{(k)}\) is the element at position \((i, j)\) in the integration matrix \(\hat{I}^{(k)}\).

So far, we can use \(\left|X^{(k)}\left(\hat{\xi}_i^{(k)}\right) - \hat{X}^{(k)}\left(\hat{\xi}_i^{(k)}\right)\right|\) to approximate the error between the numerical solution \(X^{(k)}(\tau)\) and the analytical solution \(x^{(k)}(\tau)\) of problem \(B\), which is expressed as follows

\[
\varepsilon_{\max}^{(k)}(\hat{\xi}_i^{(k)}) = \max_{l=1, \ldots, n_x} \max_{i=1, \ldots, \Gamma_k+1} \left|X_l^{(k)}\left(\hat{\xi}_i^{(k)}\right) - \hat{X}^{(k)}_l\left(\hat{\xi}_i^{(k)}\right)\right| (21)
\]

Where \(X_l^{(k)}\) and \(\hat{X}_l^{(k)}\) represent the \(l\) th element of vectors \(X^{(k)}\) and \(\hat{X}^{(k)}\), respectively.

### 4.2 Refining the mesh

On the mesh interval \(S_k = [T_{k-1}, T_k]\), we use \(\varepsilon_{\max}^{(k)}\) calculated in Equation (21) to represent the error of numerical solution \(\left(X_j^{(k)}, U_j^{(k)}\right)\). Set \(\epsilon\) is the accuracy tolerance range. If \(\varepsilon_{\max}^{(k)} \leq \epsilon\) holds for all mesh interval \(S_k, k = 1, \ldots, K\), then the numerical solution obtained is considered to be satisfactory and the iterative solving is stopped. On the contrary, it is necessary to refine the original mesh and continue the iterative solving. The detailed mesh refinement method is as follows.

#### 4.2.1 The mesh refinement in advance based on control sudden changes

For many optimal control problems, there are some sudden changes on the control curve, such as bang-bang control, switching control and so on. When hp pseudospectral method is used to solve problem \(B^N\), the sudden changes of control
Optimization results of system (22)

In order to find out the optimal control $u^*(t)$ to minimize the cost function $J$, the adaptive hp method in reference 21 is used to solve Equation (22). The results are as Figure 1.

According to the optimal control theory, this example is bang-bang control, and its analytical solution is

$$u^*(t) = \begin{cases} -1, & 0 \leq t \leq 2.22 \\ +1, & 2.22 \leq t \leq 3.44 \end{cases} \quad (23)$$

By comparing Equation (23) with Figure 1A, when the adaptive hp method\textsuperscript{21} is used to solve this example, the optimization result is in good agreement with the analytical solution. On the other hand, in the process of solving the example, in order to achieve the desired approximation accuracy, dense mesh points and collocations are needed at the sudden change point ($t = 2.22s$) of the control curve, which greatly increases the number of iterations and reduces the efficiency of the solving (see Figure 1B).

In order to effectively solve the above problems, in the improved adaptive hp method, we refine the mesh in advance based on the sudden change points of control. The basic idea is to accurately locate the sudden change points, and then refine the mesh around the points first to improve the efficiency.

Firstly, in order to locate the sudden change points, take time derivative along $u^{(k)}(t)$ at $t_i$ ($i = 1, 2, \ldots, N_k$)

$$\ddot{u}^{(k)}(t_i) = \frac{u^{(k)}(t_{i+1}) - u^{(k)}(t_i)}{t_{i+1} - t_i} \quad (24)$$

Where $t_i = (\tau_i + 1) h_k / 2 + T_{k-1}$, Equation (24) can be further written as

$$\ddot{u}^{(k)}(t_i) = \frac{2}{h_k} \cdot \frac{u^{(k)}(\tau_{i+1}) - u^{(k)}(\tau_i)}{\tau_{i+1} - \tau_i} = \frac{2}{h_k} \cdot \frac{U^{(k)}_{i+1} - U^{(k)}_i}{\tau_{i+1} - \tau_i} \quad (25)$$
Then, according to the derivative of the control and the set threshold, the interval of the control changing too fast is selected. Suppose that the derivative of control at each collocation point in interval $S_k$ is as shown in Figure 2.

In Figure 2, $\eta$ is the set threshold. When the derivative of the control exceeds $\eta$ (such as $t_\alpha$ and $[t_{\beta-1}, t_\beta, t_{\beta+1}]$ in Figure 2), we determine that there may be control sudden change points in this area, so we need to refine the mesh in advance.

Finally, mesh points are added in the area where the control changes sharply, so as to refine the grid in this area first. Taking Figure 2 as an example, we will add new mesh points $t_{\alpha-1}, t_{\alpha+1}, t_{\beta-2}, t_{\beta+2}$ to further refine the grid of areas $[t_{\alpha-1}, t_{\alpha+1}]$ and $[t_{\beta-2}, t_{\beta+2}]$. In order to describe the subsequent mesh refinement process, the new mesh points are denoted as $\{T'_k\}_{k=0}^{K'}$, and the new mesh intervals are denoted as $S'_k = [T'_{k-1}, T'_k], k = 1, 2, \ldots, K'$.

### 4.2.2 Criterion of mesh classification refinement based on state curvature

When using hp method to solve the optimal control problem, the mesh interval should be classified first and then refined.20–23 The state in some mesh intervals change dramatically, so the segmented method is suitable for this kind of interval, that is, h method. While for those mesh intervals where the state change smoothly, it is suitable to use the method of adding collocation points to refine, that is, p method. In the improved adaptive hp method, we take the curvature of state as the criterion of classification.

The set of collocation points in the new mesh interval $S'_k$ is denoted as $\Omega_k$, and the state at each collocation point is expressed as $\Omega_{\epsilon(t)}$. Based on $\Omega_k$ and $\Omega_{\epsilon(t)}$, $N$ points with equal time intervals are selected in the interval $[T'_{k-1}, T'_k]$ for Lagrange interpolation, and the coordinates of the interpolation points are recorded as $(\chi_i^{(k)}, y_i^{(k)}), i = 0, 1, \ldots, N - 1$, where $\chi_i^{(k)} = T'_{k-1} + i \cdot (T'_k - T'_{k-1}) / (N - 1)$. The curvature of each point can be calculated by using the interpolation point coordinates

$$\kappa_i^{(k)} = \frac{|\ddot{y}_i^{(k)}|}{1 + (\dot{y}_i^{(k)})^2}$$

Combined with the curvature, we set the criterion of classification refinement as

$$\text{Cri}^{(k)} = \max_{i=0,1,\ldots,N-1} \kappa_i^{(k)}$$

$$\sum_{i=0}^{N-1} \frac{\kappa_i^{(k)}}{N}$$
If the curvature threshold parameter is set to $\rho$ and $\text{Cri}^{(k)} > \rho$ is true, it can be considered that the state in the mesh interval changes too fast and need to be refined by segmentation. On the contrary, if $\text{Cri}^{(k)} \leq \rho$, it is necessary to refine the mesh by adding collocation points.

### 4.2.3 Increasing collocation points based on maximum approximation error

If it is determined in section 4.2.2 that the number of collocations needs to be further increase in mesh interval $S'_k$, here we will increase the number of collocations according to the maximum approximation error $e_{\text{max}}^{(k)}$ in Equation (21) in the following way

$$N'_k = N_k + \left[ \log_{10} \left( \frac{\epsilon^{(k)}_{\text{max}}}{\epsilon} \right) \right]$$

(28)

Where, $N_k$ is the original allocation number of interval $S'_k$, $N'_k$ is the increased allocation number, and $\epsilon$ is the set accuracy tolerance range. $[\cdot]$ represents the least integer function.

The rationality of this method is that if the solution error of the mesh interval is large, more points need to be added to improve the accuracy of the solution. If the error is small, a small number of collocations is added to ensure the solving speed.

### 4.2.4 The segmentation method based on cumulative sum of curvature

When the traditional adaptive hp method performs grid segmentation, the mesh interval that needs to be segmented is usually equally divided into several sub-intervals. This approach does not maximize the efficiency of segmentation, that is, it does not allocate more mesh points to those locations that need more refinement. In order to further improve the efficiency of mesh refinement, we design a segmentation method based on the cumulative sum of curvature, that is, assigning more mesh points to the position where the curvature of the state curve is larger.

Firstly, the number of subintervals to be added is estimated. The principle is that if the approximation error of the solution in the mesh interval is large, the interval needs to be divided into more subintervals. On the contrary, the number of subintervals is less. According to this principle, the number of sub-intervals divided in mesh interval $S'_k$ is determined by the following formula

$$M^{(k)} = \max \left( 2, \left[ \sigma \cdot \log_{10} \left( \frac{e_{\text{max}}^{(k)}}{\epsilon} \right) \right] \right)$$

(29)

Where $\max(\circ, \ast)$ is the maximum of $\circ$ and $\ast$. $\sigma$ is the set product factor.

Then, according to the curvature of the state curve in mesh interval $S'_k$, the position of the new mesh point is determined. It is known from Equation (29) that interval $S'_k$ needs to be divided into $M^{(k)}$ subintervals, that is, $M^{(k)} - 1$ mesh points need to be added between $T'_{k-1}$ and $T'_k$. In order to determine the location of these mesh points, we need to refer to the curvature $k_i^{(k)}$ at each interpolation point $\left( x_i^{(k)}, y_i^{(k)} \right)$ in Equation (26) to ensure that more dense mesh points are configured at the position where the state changes rapidly. Firstly, the continuous curvature curve is taken as an example to analyze, assuming that the curvature changes with time as shown in the figure below. According to the above mesh segmentation principle, if the original interval needs to be divided into four sub-intervals (as shown in Figure 3), we can select new mesh points $m_1$, $m_2$ and $m_3$ such that $S_1 = S_2 = S_3 = S_4 = S/4$, where $S_1$, $S_2$, $S_3$ and $S_4$ represent the area bounded by the curvature curve and the time axis in sub-intervals $[T'_{k-1}, m_1]$, $[m_1, m_2]$, $[m_2, m_3]$ and $[m_3, T'_k]$ respectively, and $S$ represents the total area bounded by the curvature curve and the time axis in the whole interval $[T'_{k-1}, T'_k]$.

Referring to the above analysis, for the discrete curvature, the mesh point configuration rule of the improved adaptive hp method in this paper is to select $M^{(k)} - 1$ points from the $N$ interpolation points $\left( x_i^{(k)} \right)$ in the interval $S'_k$, and record them as $m_i^{(k)}$, $i = 1, \ldots, M^{(k)} - 1$, so that the sum of the curvatures at the interpolation points in each sub
interval $[m_{i-1}^{(k)}, m_i^{(k)}]$ is approximately equal. According to this rule, the algorithm for selecting new mesh points is as follows

$$m_i^{(k)} = \left\{ \begin{array}{ll}
\kappa_j^{(k)} & \sum_{r=0}^{i} \kappa_r^{(k)} \geq i \cdot \bar{S}, \quad \sum_{r=0}^{i-1} \kappa_r^{(k)} < i \cdot \bar{S} \\
\end{array} \right. \quad (30)$$

Where $\bar{S} = \sum_{i=0}^{N-1} \kappa_i^{(k)} / M^{(k)}$.

### 4.2.5 Algorithm procedure

So far, the key steps of the improved adaptive hp method proposed in this paper have been described, and the process of its core algorithm is summarized as follows

1. **Step 1**: Select the initial mesh points $[T_0, T_1, \ldots, T_K]$, and set the initial collocation points $\left( r_1^{(k)}, \ldots, r_N^{(k)} \right)$ in each mesh interval $S_k = [T_{k-1}, T_k]$.
2. **Step 2**: Based on the current mesh points and collocation points, use the NLP solver to solve the problem $B^N$, and get the numerical solution $(X_j^{(k)}, U_j^{(k)})$.
3. **Step 3**: Use the method in section 4.1 to approximate the error between the numerical solution and the analytical solution, and compare the maximum error $e_{max}^{(k)}$ in each mesh interval $S_k$ with the accuracy tolerance $\varepsilon$ to determine whether the original grid needs further refinement. If $e_{max}^{(k)} \leq \varepsilon$ is established in all mesh intervals, the existing numerical solution can be considered as a satisfactory solution, and skip to Step 4. Otherwise, skip to Step 5.
4. **Step 4**: Complete the iteration of the solution, and take the current numerical solution as the final output.
5. **Step 5**: The existing numerical solutions do not meet the accuracy requirements, so the mesh needs to be further refined:
   i. Determine the location of the sudden change points based on the change rate of control, add new mesh points around them, and refine the original mesh points into $[T'_{0}, T'_{1}, \ldots, T'_{K'}]$ in advance.
   ii. Calculate the curvature $\kappa_i^{(k)}$ of the state $X_i^{(k)}$ in each mesh interval $S'_k = [T'_{k-1}, T'_{k}]$, so as to continue to mesh classification Refinement. If $Cr_i^{(k)}$ in Equation (27) is greater than the threshold $\rho$, go to step iii. Otherwise, go to step iv.
   iii. According to the method in section 4.2.4, the mesh interval $S'_k$ is segmented based on curvature accumulation.
   iv. Based on the maximum approximation error $e_{max}'^{(k)}$ in Equation (21), as described in 4.2.3, add a certain number of collocation points in the mesh interval $S'_k$. 

---

**FIGURE 3** Mesh interval segmentation
1. Step 6: The refined mesh points and collocation points in step 5 are brought back to step 2 for a new round of solving.

The procedure of this method is shown in Figure 4.

5  |  CONVERGENCE ANALYSIS

In order to prove the convergence of the algorithm, the following assumptions are made for problem $B$ (Equation (1)).

**Assumption 1.** For $l \geq 3$, in the domain $C^{l+1}(\mathbb{R}^{n_x}) \times L^\infty(\mathbb{R}^{n_u}) \times C^{l+1}(\mathbb{R}^{n_u}) \times C^{l+1}(\mathbb{R}^{n_x}) \times \mathbb{R}^{n_h}$, there is an optimal solution $(x^*, u^*, \lambda^*, \mu^*, \nu^*)$ for problem $B$ (Equation [1]) so that the Pontryagin minimum condition (Equation [2]) holds. At the same time, there is $\theta > 0$ and open set $\Omega \subset \mathbb{R}^{n_x} \times \mathbb{R}^{n_u}$ such that $B_\theta(x^*(t), u^*(t)) \subset \Omega$ holds for any $t \in [t_0, t_f]$, and the first two derivatives of functions $F(\cdot), f(\cdot)$ and $h(\cdot)$ are Lipschitz continuous in $\Omega$, and the first two derivatives of $E(\cdot)$ and $e(\cdot)$ are Lipschitz continuous in $B_\theta(x^*(t_f))$.

**Assumption 2.** For any $t \in [t_0, t_f]$, consider the following matrices

$$A = \nabla_{xx} E(x^*(t_f))$$

$$B = \nabla_{xx} e(x^*(t_f))$$

$$C = \begin{pmatrix} Q(t) & S(t) \\ S^T(t) & R(t) \end{pmatrix}$$
Where $Q(t) = \nabla_x H (x^*(t), u^*(t), \lambda^*(t), \mu^*(t)), S(t) = \nabla_u H (x^*(t), u^*(t), \lambda^*(t), \mu^*(t)), R(t) = \nabla_v H (x^*(t), u^*(t), \lambda^*(t), \mu^*(t))$. Assume that the smallest eigenvalues of matrices $A, B,$ and $C$ are strictly greater than zero.

**Assumption 3.** For any $t \in [t_0, t_f]$, there is $\|\nabla_x f(x^*(t), u^*(t))\| \leq \min_{k=1}^{K} \frac{1}{4h_k}$.

Next, analyze the convergence of the improved adaptive hp algorithm proposed in this paper, i.e. to prove that for the NLP $B^N$ (Equation [15]) there is a minimum point $(x^{(k)}, u^{(k)})$ and Lagrangian multiplier vectors $\lambda^{(k)}_i, \mu^{(k)}_i, \nu^{(k)}_i$ such that $(x^{(k)}_i, u^{(k)}_i, \lambda^{(k)}_i, \mu^{(k)}_i, \nu^{(k)}_i)$ converges to the optimal solution $(x^{(k)}_i, u^{(k)}_i, \lambda^{(k)}_i, \mu^{(k)}_i, \nu^{(k)}_i)$ of problem $B$, where $k = 1, \ldots, K, i = 1, \ldots, N_k, x^{(k)}_i = x^{(k)}(t_i), t^{(k)}_i = u^{(k)}(t_i), \lambda^{(k)}_i = \lambda^{(k)}(t_i), \mu^{(k)}_i = \mu^{(k)}(t_i)$.

**Theorem 1.** If Assumption 1~3 hold, then for problem $B^N$ (Equation [15]), there is a minimum point $(x^{(k)}_i, u^{(k)}_i)$ and Lagrangian multiplier vectors $\lambda^{(k)}_i, \mu^{(k)}_i,$ and $\nu^{(k)}_i$ satisfy

$$
\max \left\{ \left\| X_{N_k} - x^* \right\|_\infty, \left\| U_{N_k} - u^* \right\|_\infty, \left\| \lambda_{N_k} - \lambda^* \right\|_\infty, \left\| \mu_{N_k} - \mu^* \right\|_\infty, \left\| \nu_{N_k} - \nu^* \right\|_\infty \right\} \leq \frac{C_k h_k^{b+1}}{N_k^{-\frac{1}{2}}} \tag{31}
$$

Where $X_{N_k} = (x^{(k)}_1, \ldots, x^{(k)}_{N_k})^T, x^*(k) = (x^{(k)}_1, \ldots, x^{(k)}_{N_k})^T, U_{N_k} = (U^{(k)}_1, \ldots, U^{(k)}_{N_k})^T, U^*(k) = (U^{(k)}_1, \ldots, U^{(k)}_{N_k})^T, \lambda_{N_k} = (\lambda^{(k)}_1, \ldots, \lambda^{(k)}_{N_k})^T, \lambda^*(k) = (\lambda^{(k)}_1, \ldots, \lambda^{(k)}_{N_k})^T, \mu_{N_k} = (\mu^{(k)}_1, \ldots, \mu^{(k)}_{N_k})^T, \mu^*(k) = (\mu^{(k)}_1, \ldots, \mu^{(k)}_{N_k})^T$. In addition, $k = 1, \ldots, K, N_k$ is the number of LGR points in the mesh interval $S_k$, $h_k$ represents the interval length of $S_k$, $C_k$ is a constant greater than 0, and the definition of $l_k$ is similar to the definition of $l$ in Assumption 1.

**Proof.** First, consider the first-order necessary conditions for the minimum point of problem $B^N$ (Equation [15]).

Introduce Lagrangian multiplier vectors $\Lambda^{(k)}_i \in \mathbb{R}^m, m^{(k)} \in \mathbb{R}^m$ and $g^{(k)} \in \mathbb{R}^m$ (all three are assumed to be row vectors), and construct the following Lagrangian function

$$
L \left( x^{(k)}_i, u^{(k)}_i, \lambda^{(k)}_i, m^{(k)}_i, g^{(k)}_i \right) = E \left( x^{(1)}_i, x^{(K)}_{N_k+1} \right) + \sum_{k=1}^{K} \sum_{j=1}^{N_k} \frac{h_k}{2} \omega^{(k)}_j F \left( x^{(k)}_j, u^{(k)}_j \right) + \sum_{k=1}^{K} \sum_{j=1}^{N_k} \lambda^{(k)}_j \left( h_k f \left( x^{(k)}_j, u^{(k)}_j \right) - \sum_{j=1}^{N_k} x^{(k)}_j D^{(k)}_{ij} \right) + \sum_{k=1}^{K} \sum_{j=1}^{N_k} m^{(k)}_j \cdot h^T \left( x^{(k)}_j, u^{(k)}_j \right) \tag{32}
$$

Let $D^{(k)}_{ij}$ denote the element in column $i$ of differentiation matrix $D^{(k)} = \left[ D^{(k)}_{ij} \right]$. At the same time $A^{(k)} = \left( A^{(k)}_1, \ldots, A^{(k)}_{N_k} \right)^T$. Calculate the gradient of Lagrangian function $L$ in Equation (32) to $x^{(K)}_{N_k+1}, x^{(1)}_j, x^{(1)}_j$ and $u^{(1)}_j$, respectively. The first-order necessary conditions for the minimum point are as follows

1. $\nabla_{x^{(k)}_{N_k+1}} L = 0$

$$
\nabla_{x^{(k)}_{N_k+1}} \left( E \left( x^{(1)}_1, x^{(K)}_{N_k+1} \right) + g^{(k)}_i \cdot e^T \left( x^{(1)}_1, x^{(K)}_{N_k+1} \right) \right) - \left( D^{(k)}_{N_k+1} \right)^T \cdot A^{(k)} = 0 \tag{33}
$$

2. $\nabla_{x^{(1)}_1} L = 0$

$$
\nabla_{x^{(1)}_1} \left( E \left( x^{(1)}_1, x^{(K)}_{N_k+1} \right) + g^{(k)}_i \cdot e^T \left( x^{(1)}_1, x^{(K)}_{N_k+1} \right) \right) + \frac{h_1}{2} \omega^{(1)}_1 \nabla_{x^{(1)}_1} E \left( x^{(1)}_1, u^{(1)}_1 \right) + \frac{h_1}{2} \omega^{(1)}_1 \nabla_{x^{(1)}_1} \mathbf{f} \left( x^{(1)}_1, u^{(1)}_1 \right) \tag{34}
$$
The proof of Theorem 1 requires the following Proposition.\(^{31}\)

Combining Equations (37–40), rewrite the first-order necessary conditions (33) \(~\sim\) (36) into the following form

1. \(\nabla_{\bm{X}^{(K)}} L = 0\)

\[
\nabla_{\bm{X}^{(K)}} \left( E \left( \bm{X}^{(1)}, \bm{X}^{(K)}_{N_{k}+1} \right) + \nu^{(k)} \cdot \bm{e}^T (\bm{X}^{(1)}_{N_{k}}, \bm{X}^{(K)}_{N_{k}+1}) \right) - \lambda^{(K)}_{N_{k}+1} = 0
\]

2. \(\nabla_{\bm{X}_{i}} L = 0\)

\[
\sum_{j=1}^{N_i} \nabla_{\bm{X}_{i}^{(1)}} \lambda_{j}^{(1)} + \frac{h_k}{2} \nabla_{\bm{X}_{i}^{(1)}} \left( F \left( \bm{X}_{i}^{(1)}, \bm{U}_{i}^{(1)} \right) + \lambda_{j}^{(1)} \bm{f}^T \left( \bm{X}_{i}^{(1)}, \bm{U}_{i}^{(1)} \right) \right) + \mu_{1}^{(1)} \nabla_{\bm{X}_{i}^{(1)}} \bm{h}^T (\bm{X}_{i}^{(1)}, \bm{U}_{i}^{(1)}) + \frac{1}{\omega_{j}^{(1)}} \nabla_{\bm{X}_{i}^{(1)}} \left( E (\bm{X}^{(1)}_{1}, \bm{X}^{(K)}_{N_{k}+1}) + \nu^{(k)} \cdot \bm{e}^T (\bm{X}^{(1)}_{N_{k}}, \bm{X}^{(K)}_{N_{k}+1}) \right) - \lambda_{1}^{(1)} = 0
\]

3. \(\nabla_{\bm{X}_{i}^{(k)}} L = 0\)

\[
\sum_{j=1}^{N_i} \nabla_{\bm{X}_{i}^{(k)}} \lambda_{j}^{(k)} + \frac{h_k}{2} \nabla_{\bm{X}_{i}^{(k)}} \left( F \left( \bm{X}_{i}^{(k)}, \bm{U}_{i}^{(k)} \right) + \lambda_{j}^{(k)} \bm{f}^T \left( \bm{X}_{i}^{(k)}, \bm{U}_{i}^{(k)} \right) \right) + \mu_{1}^{(k)} \nabla_{\bm{X}_{i}^{(k)}} \bm{h}^T (\bm{X}_{i}^{(k)}, \bm{U}_{i}^{(k)}) = 0 \quad (1 \leq k \leq K, 1 \leq i \leq N_k, \bm{X}^{(k)}_{i} \neq \bm{X}^{(1)}_{i})
\]

4. \(\nabla_{\bm{U}_{i}^{(k)}} L = 0\)

\[
\frac{h_k}{2} \nabla_{\bm{U}_{i}^{(k)}} \left( F \left( \bm{X}_{i}^{(k)}, \bm{U}_{i}^{(k)} \right) + \lambda_{j}^{(k)} \bm{f}^T \left( \bm{X}_{i}^{(k)}, \bm{U}_{i}^{(k)} \right) \right) + \mu_{1}^{(k)} \nabla_{\bm{U}_{i}^{(k)}} \bm{h}^T (\bm{X}_{i}^{(k)}, \bm{U}_{i}^{(k)}) = 0 \quad (1 \leq k \leq K, 1 \leq i \leq N_k)
\]
Proposition 1. If Assumption 1 ~ Assumption 3 are established. For \( \theta^* \in \Xi \) and \( r > 0 \), let the mapping \( \Gamma : \Xi \mapsto \Psi \) be continuously Frechet differentiable within \( B_r(\theta^*) \). Where \( \Xi \) is a Banach space and \( \Psi \) is a normed linear space. If the following conditions are true.

(C1) \( \forall \Gamma(\theta^*) \) is invertible.

(C2) For any \( \sigma > 0 \), there is \( r > 0 \) such that \( \lVert \nabla \Gamma(\theta) - \nabla \Gamma(\theta^*) \rVert \leq \sigma \) holds for all \( \theta \in B_r(\theta^*) \).

Let \( \sigma = \lVert \nabla \Gamma(\theta^*)^{-1} \rVert \), if there is \( \sigma \sigma < 1 \) and \( \lVert \Gamma(\theta^*) \rVert \leq (1 - \sigma \sigma) r / \sigma \). Then there is unique \( \theta \in B_r(\theta^*) \) such that \( \Gamma(\theta) = 0 \), and at the same time there is

\[
\lVert \theta - \theta^* \rVert_{\infty} \leq \frac{\sigma \sigma}{1 - (1 - \sigma \sigma) r / \sigma} \lVert \Gamma(\theta^*) \rVert_{\infty}
\]

Let \( \theta = (X_N^{e}, U_N^{e}, \lambda_N^{e}, \mu_N^{e}, \nu^{(k)}) \) and \( \theta^* = (X^{e(k)}, U^{e(k)}, \lambda^{e(k)}, \mu^{e(k)}, \nu^{(k)}) \). At the same time, combining Equations (15) and (41) ~ (44), the expression of mapping \( \Gamma \) at \( \theta^* \) is defined as \( \Gamma(\theta^*) = (\Gamma_1(\theta^*) \, \ldots \, \Gamma_9(\theta^*))^T \), where

\[
\Gamma_1(\theta^*) = X_{N_{k+1}}^{e} \left( E + \nu^{(k)} \cdot e^T \right) - \lambda_{N_{k+1}}^{e(k)}
\]

\[
\Gamma_2(\theta^*) = \sum_{j=1}^{N_k} D_{ij}^{e(1)} \lambda_{ij}^{e(1)} + \frac{h_1}{2} \nabla X_{ij}^{e(1)} \left( F + \lambda_{ij}^{e(1)} F^T \right)
\]

\[
+ \mu_{ij}^{e(1)} \nabla X_{ij}^{e(1)} \left( F + \lambda_{ij}^{e(1)} F^T \right)
\]

\[
\Gamma_3(\theta^*) = \sum_{j=1}^{N_k} D_{ij}^{e(k)} \lambda_{ij}^{e(k)} + \frac{h_2}{2} \nabla X_{ij}^{e(k)} \left( F + \lambda_{ij}^{e(k)} F^T \right)
\]

\[
+ \mu_{ij}^{e(k)} \nabla X_{ij}^{e(k)} \left( F + \lambda_{ij}^{e(k)} F^T \right)
\]

\[
\Gamma_4(\theta^*) = \frac{h_k}{2} \nabla U_{i}^{e(k)} \left( F + \lambda_{i}^{e(k)} F^T \right) + \mu_{i}^{e(k)} \nabla U_{i}^{e(k)} h^T (1 \leq k \leq K, 1 \leq i \leq N_k)
\]

\[
\Gamma_5(\theta^*) = \mu_{i}^{e(k)} h^T \left( X_{i}^{e(k)}, U_{i}^{e(k)} \right) (1 \leq k \leq K, 1 \leq i \leq N_k)
\]

\[
\Gamma_6(\theta^*) = \sum_{j=1}^{N_{k+1}} X_{ij}^{e(k)} D_{ij}^{e(k)} - \frac{h_k}{2} f \left( X_{ij}^{e(k)}, U_{i}^{e(k)} \right) \quad i = 1, 2, \ldots, N_{k+1} \quad k = 1, 2, \ldots, K
\]

\[
\Gamma_7(\theta^*) = e \left( X_{1}^{e(1)}, X_{N_{k+1}}^{e(K)} \right)
\]

\[
\Gamma_8(\theta^*) = h \left( X_{i}^{e(k)}, U_{i}^{e(k)} \right)
\]

\[
\Gamma_9(\theta^*) = X_{N_{k+1}}^{e(k)} - X_{1}^{e(k+1)}
\]

From reference 52, we know that the conditions C1 and C2 in Proposition 1 are established. Next, consider the size of \( \lVert \Gamma(\theta^*) \rVert_{\infty} \).

According to Assumption 1, \( \theta^* \) satisfies Equations (1) and (2), so for \( \Gamma(\theta^*) \) defined above, there is

\[
\Gamma_1(\theta^*) = \Gamma_4(\theta^*) = \Gamma_5(\theta^*) = \Gamma_7(\theta^*) = \Gamma_9(\theta^*) = 0
\]

\[
\Gamma_8(\theta^*) \leq 0
\]

Next, in order to continue to examine the size of \( \lVert \Gamma_2(\theta^*) \rVert_{\infty}, \lVert \Gamma_3(\theta^*) \rVert_{\infty} \) and \( \lVert \Gamma_6(\theta^*) \rVert_{\infty} \), first introduce the following lemma. 52

Lemma 1. If the Lebesgue constant \( \ell_N \) is defined as

\[
\ell_N = \max_{t \in [-1,1]} \sum_{j=1}^{N+1} |f_j(t)|
\]
Here, the definition of Lagrange interpolation polynomial \( l_i(\tau) \) is shown in Equation (16). Then there is \( \ell_N = O(\log N) \).

**Lemma 2.** If the Lebesgue constant \( \ell_{N-1} \) is defined as

\[
\ell_{N-1} = \max_{\tau \in [-1,1]} \left| \sum_{i=1}^{N} l_i(\tau) \right| \tag{57}
\]

The definition of Lagrange interpolation polynomial \( l_i(\tau) \) is shown in Equation (17). Then there is, \( \ell_{N-1} = O\left(N^{\frac{1}{2}}\right) \).

**Lemma 3.** For any \( y \in C^1[-1,1] \), define the \( N \) order polynomial \( y^N(t) = \sum_{i=1}^{N+1} y(t_i) l_i(t) \), where the Lagrange polynomial \( l_i(t) \) is defined as shown in Equation (16), then

\[
\| y - y^N \|_\infty \leq (1 + 2N^2\ell_N) \inf_{q \in \mathcal{P}_{N+1}} \| y - q \|_\infty \tag{58}
\]

Where, the definition of Lebesgue constant \( \ell_N \) is shown in Equation (56).

**Lemma 4.** If \( y(t) \in C^l \) and \( N > l \geq 0 \), then there is

\[
\inf_{q \in \mathcal{P}_N} \| y - q \|_\infty \leq c_l \left\| \frac{d^l y}{dt^l} \right\|_\infty \tag{59}
\]

Where \( c_l = 2 \times 6^{l+1}e(1 + b)^{-1} \).

**Lemma 5.** According to the definition in Equations (39) and (40), matrix \( D^{(k)} = \begin{bmatrix} D_{ij}^{(k)} \end{bmatrix}_{N_k \times N_k} \) is a differentiation matrix. That is, if \( p \) is a polynomial of degree \( N_k - 1 \) and the \( i \)th element of vector \( P \in \mathbb{R}^{N_k} \) is \( P_i = p_i = p(\tau_i) \), then there is

\[
\left( D^{(k)} P \right)_i = \dot{p}(\tau_i), \quad 1 \leq i \leq N_k \tag{60}
\]

The proof of Lemma 1 ~ 5 can be found in reference 52.

In order to analyze the size of \( \left\| \Gamma_3(\theta^*) \right\|_\infty \), combining with Equation (2), we can know that in Equation (48),

\[
\mu_i^{(k)} \nabla_{X_i}^{\omega_i} h^T = 0, \quad \frac{h_i}{2} \nabla_{X_i}^{\omega_i} \left( F + \lambda_i^{(k)} F^T \right) = -\lambda_i^{(k)}(\tau_i). \tag{61}
\]

From Lemma 5,

\[
\sum_{j=1}^{N_k} D_{ij}^{(k)} \lambda_j^{(k)} = \left[ \lambda_i^{(k)}(\tau_i) \right]^{N_k-1} \tag{62}
\]

Where, the superscript \( N_k - 1 \) denotes that \( \lambda_i^{(k)} \) is a polynomial of order \( N_k - 1 \) obtained by Lagrange interpolation. Combining Equations (61) and (62), there is

\[
\Gamma_3(\theta^*) = \left[ \lambda_i^{(k)}(\tau_i) \right]^{N_k-1} - \lambda_i^{(k)}(\tau_i) \tag{63}
\]

Let \( \left[ \lambda_i^{(k)}(\tau_i) \right]^{N_k-1}_j \) denote the \( j \)th element of vector \( \left[ \lambda_i^{(k)}(\tau_i) \right]^{N_k-1} \), and \( \lambda_j^{(k)}(\tau_i) \) denote the \( j \)th element of \( \lambda_i^{(k)}(\tau_i) \). Therefore, Equation (63) can be simplified to

\[
\left| \lambda_i^{(k)}(\tau_i) \right|^{N_k-1}_j - \lambda_j^{(k)}(\tau_i) \leq \left\| \lambda_i^{(k)} \right\|_j^{N_k-1} - \lambda_j^{(k)} \|_\infty \tag{64}
\]
According to Lemmas 2, 3, and 4, we can get

\[
\| \lambda^{*}(k) \|_j^{N_k-1} - \lambda_j^{*}(k) \|_{\infty} \leq \left( 1 + 2(N_k - 1)^2 O \left( \frac{1}{N_k^2} \right) \right) \inf_{q \in P_{N_k-1}} \| \lambda_j^{*}(k) - q \|_{\infty} \\
\leq \left( 1 + 2(N_k - 1)^2 O \left( \frac{1}{N_k^2} \right) \right) \frac{c_k}{N_k^{\alpha_1}} \left\| \frac{d^{k+1} \lambda_j^{*}(k)}{dt^{k+1}} \right\|_{\infty} \\
\leq \frac{c_k h_k^{l+1}}{N_k^{\alpha_2-\frac{1}{2}}} 
\]

(65)

Since Equations (64) and (65) hold for any \( i \) and \( j \), there is

\[
\| \Gamma_3 (\theta^*) \|_{\infty} \leq \frac{c_k h_k^{l+1}}{N_k^{\alpha_2-\frac{1}{2}}} 
\]

(66)

And for \( \| \Gamma_2 (\theta^*) \|_{\infty} \), according to the similar analysis method, we can also get

\[
\| \Gamma_2 (\theta^*) \|_{\infty} \leq \frac{c_k h_k^{l+1}}{N_k^{\alpha_2-\frac{1}{2}}} 
\]

(67)

Finally, in order to analyze the size of \( \| \Gamma_8 (\theta^*) \|_{\infty} \), we know \( \frac{h_k}{2} f(X^{*}(k), U^{*}(k)) = \dot{x}^{*}(k) (\tau_i) \) according to Equation (1), and because \( D^{(k)} = [D^{(k)}_{ij}] \) is a differentiation matrix, Equation (51) can be written as

\[
\Gamma_8 (\theta^*) = [\dot{x}^{*}(k) (\tau_i)]^{N_k} - \dot{x}^{*}(k) (\tau_i) 
\]

(68)

Where, the superscript \( N_k \) denotes that \( x^{*}(k) \) is a polynomial of order \( N_k \) obtained by Lagrange interpolation. Let \( [\dot{x}^{*}(k) (\tau_i)]_j^{N_k} \) denote the \( j \) th element of \( [x^{*}(k) (\tau_i)]^{N_k} \), and \( \dot{x}^{*}(k) (\tau_i) \) denote the \( j \) th element of \( x^{*}(k) (\tau_i) \). There is,

\[
\left| [\dot{x}^{*}(k) (\tau_i)]_j^{N_k} - \dot{x}^{*}(k) (\tau_i) \right| \leq \left\| [\dot{x}^{*}(k)]_j^{N_k} - \dot{x}^{*}(k) \right\|_{\infty} 
\]

(69)

Combined with Lemma 2, 3, and 4, we can get

\[
\| [x^{*}(k)]_j^{N_k} - \dot{x}^{*}(k) \|_{\infty} \leq \left( 1 + 2N_k^2 O(\log N) \right) \inf_{q \in P_{N_k-1}} \| x_j^{*}(k) - q \|_{\infty} \\
\leq \left( 1 + 2N_k^2 O(\log N) \right) \frac{c_k}{N_k^{\alpha_1}} \left\| \frac{d^{k+1} x^{*}(k)}{dt^{k+1}} \right\|_{\infty} \\
\leq \frac{c_k \log N}{N_k^{\alpha_2-\frac{1}{2}}} \left( \frac{h_k}{2} \right)^{l+1} \left\| \frac{d^{k+1} x^{*}(k)}{dt^{k+1}} \right\|_{\infty} \\
\leq \frac{c_k h_k^{l+1} \log N}{N_k^{\alpha_2-\frac{1}{2}}} \leq \frac{c_k h_k^{l+1}}{N_k^{\alpha_2-\frac{1}{2}}} 
\]

(70)
Because Equations (69) and (70) hold for any i and j, we get

\[ \left\| \Gamma_\theta (\theta^*) \right\|_\infty \leq \frac{c_i h_k^{i+1}}{N_k^{i-\frac{3}{2}}} \]  

(71)

Combining Equations (55), (66), (67) and (71), we can see that

\[ \left\| \Gamma (\theta^*) \right\|_\infty \leq \frac{c_i h_k^{i+1}}{N_k^{i-\frac{3}{2}}} \]  

(72)

For \( \sigma = \left\| \nabla \Gamma (\theta^*) \right\|^{-1} \), \( \sigma \sigma < 1 \) can be established by taking a small enough \( \sigma \), in addition, according to Equation (72), \( \left\| \Gamma (\theta^*) \right\| \leq (1 - \sigma \sigma) r/\sigma \) can be established by taking a large enough \( N_k \). Therefore, according to Proposition 1 and Equation (72), we know

\[
\left\| \theta - \theta^* \right\|_\infty = \max \left\{ \left\| X_{N_k} - X_s^{(k)} \right\|_\infty, \left\| U_{N_k} - U_s^{(k)} \right\|_\infty, \left\| \lambda_{N_k} - \lambda_s^{(k)} \right\|_\infty, \left\| \mu_{N_k} - \mu_s^{(k)} \right\|_\infty, \left\| v_s^{(k)} - v_s^{(k)} \right\|_\infty \right\} \\
\leq \frac{\sigma}{1 - \sigma \sigma} \left\| \Gamma (\theta^*) \right\|_\infty \\
\leq \frac{\sigma}{1 - \sigma \sigma} \frac{c_i h_k^{i+1}}{N_k^{i-\frac{3}{2}}}
\]  

(73)

Since \( \sigma/(1 - \sigma \sigma) > 0 \) is bounded, let \( C_i = \sigma c_i/(1 - \sigma \sigma) \). In conclusion, Theorem 1 is established.

6 | SIMULATION ANALYSIS

We verify the effectiveness of the improved adaptive hp algorithm through several groups of simulation examples, and illustrate the advantages of the proposed method by setting up the simulation comparison with the methods in the related research. In this part, all simulations rely on Lenovo desktop computers, whose processor is Intel(R) Core(TM) i5-4570, CPU clock speed is 3.20GHz, memory is 4.00G. And the simulation software platform is MATLAB 2014a. In addition, the algorithm proposed in this paper is developed based on the open source hp pseudospectral optimization software GPOPS, and the NLP solver used is SNOPT.

Example 1. Soft lunar landing problem.

The optimal control problem for soft landing on the moon was proposed in reference 53, and it is restated here as follows:

Find the optimal control \( u \) such that

\[
\begin{aligned}
\text{min} \quad & J = \int_0^{t_f} u \, dt \\
\text{s.t.} \quad & \dot{h} = v \\
& \dot{v} = -g + u \\
& h(0) = 10, \quad h(t_f) = 0 \\
& v(0) = -2, \quad v(t_f) = 0 \\
& 0 \leq u \leq 3
\end{aligned}
\]  

(74)

Where \( h \) is the height from the lunar surface and \( v \) is the falling velocity. \( g \) represents the gravitational constant of the lunar surface, where \( g = 1.6 \). According to the optimal control theory, problem (74) is a bang-bang control problem, and the analytical solution of the optimal control \( u^*(t) \) can be expressed as
\begin{equation}
 u^*(t) = \begin{cases}
 0 & 0 \leq t \leq t^*_s \\
 3 & t^*_s < t \leq t^*_f 
\end{cases}
\end{equation}

Combining with the initial conditions set by Equation (74), we can calculate that the switching time \( t^*_s = 1.3117 \) \( \text{s} \) and the terminal time \( t^*_f = 4.2394 \) \( \text{s} \).

At the same time, we use the improved adaptive hp algorithm proposed in this paper to solve the problem (74), and the relevant parameters in the algorithm are selected as follows: \( \epsilon = 10^{-6}, \eta = 1, \sigma = 1.2, \rho = 3 \). The initial mesh points are taken as \([-1, 1]\), and the number of initial LGR collocation points is taken as 6. In addition, under the same tolerance range \( \epsilon \) of mesh accuracy, the hp method in reference 23 and the ph-(\( N_{\min}, N_{\max} \)) method in reference 20 are used to solve the problem and compare with the method in this paper. The notation ph-(\( N_{\min}, N_{\max} \)) here represents the range of the number of collocation points of the ph method from \( N_{\min} \) to \( N_{\max} \). The simulation results are shown in Figure 5.

As shown in Figure 5A–C, all three methods can accurately solve the problem under the same tolerance range \( \epsilon \) of mesh accuracy, while the accuracy of the method in this paper is slightly higher than that of the other two methods. On the other hand, at the control sudden change point \( t = 1.3117 \) \( \text{s} \), the hp method and the ph-(3,14) method need to greatly increase the number of collocation points and the number of mesh intervals to achieve an effective solving to the original problem, which greatly increases the number of mesh iterations and sacrifices the efficiency of solving (see Figure 5E,F). As shown in Figure 5D, the improved adaptive hp method proposed in this paper can accurately locate the position of the control sudden change point, so as to achieve a higher solution accuracy with a smaller number of collocation points and iterations. In addition, from the comparison results in Table 1, it can be seen that the proposed method needs less mesh iterations (only 2 iterations), fewer collocation points and mesh intervals, which makes the method more efficient (CPU time is 0.14 s, significantly less than the other two methods).

**Example 2.** Hyper-sensitive problem.

Consider the Hyper-sensitive optimal control problem mentioned in reference 54 and restate it as follows:

\begin{equation}
\begin{aligned}
\min J &= \frac{1}{2} \int_0^{t_f} (x^2 + u^2) \; dt \\
\text{s.t.} \quad &x = -x^3 + u \\
&x(0) = 1 \\
&x(t_f) = 1 
\end{aligned}
\end{equation}

The terminal time \( t_f \) is fixed, which is taken as \( t_f = 5000 \) here. The improved adaptive hp method proposed in this paper is used to solve the problem (76), and the initial mesh points are set to \([-1, 1]\), and the initial number of LGR points is set to 6. The relevant parameters in the algorithm are set as: \( \epsilon = 10^{-6}, \eta = 1, \sigma = 1.2, \rho = 3 \). Similarly, the hp method 23 and the ph-(\( N_{\min}, N_{\max} \)) method 20 are compared with the method in this paper. The simulation results are shown in Figure 6.

It can be seen from the simulation results that the results obtained by the three methods are basically the same, and a large number of LGR points are allocated in the initial and final stages of the state curve (see Figure 6A,C,E). However, the mesh refinement process of these three methods is different: Although the ph-(3,14) method can continuously improve the solution accuracy by increasing the number of LGR points and mesh intervals, it requires a large number of LGR points and the dense grid to deal with this problem (see Figure 6F), which affects the efficiency of the solving. The reason is that when the method is performing mesh refinement, only when the number of collocation points increases to the upper limit \( N_{\max} \) will the mesh-segmented refinement begin, so that a large number of LGR points are inevitably needed at positions where the state changes rapidly. While hp method can determine whether the grid needs to be segmented refined by calculating the curvature of the state, which makes the mesh refinement process more targeted to a certain extent, reduces the number of LGR points required (see Figure 6D), and improves the efficiency of solving. The algorithm proposed in this paper improves the method of evenly dividing the grid in hp method, so that the position where the faster the state changes, the more collocations can be allocated, which further strengthens the pertinence of hp method for mesh refinement, and significantly reduces the number of mesh iterations and the total number of collocations required (see Figure 6B). Table 2 also shows the same conclusion. As shown in
FIGURE 5  Optimization results of Example 1
### Table 1 Comparison of mesh refinement results of Example 1

| Method     | CPU time (s) | Total number of collocation points | Number of mesh intervals | Number of grid iterations |
|------------|--------------|-----------------------------------|--------------------------|--------------------------|
| Improved hp | 0.14         | 48                                | 6                        | 2                        |
| hp         | 0.41         | 88                                | 7                        | 8                        |
| ph-(3,14)  | 0.25         | 127                               | 9                        | 5                        |
| ph-(5,16)  | 0.26         | 113                               | 7                        | 6                        |
| ph-(6,18)  | 0.33         | 127                               | 7                        | 7                        |

Table 2, compared with the other two methods, the improved adaptive hp method proposed in this paper requires significantly fewer collocation points and mesh iterations when solving the problem, and the solving efficiency is also higher (CPU time is 0.14 s).

### Example 3. Dynamic soaring problem.

Considering that the models of the optimal problems involved in the first two simulation examples are simple systems, in order to further test the performance of the algorithm proposed in this paper to solve the optimal problems of complex systems, the following control problems are introduced here.

Find the optimal control \((C_L, \sigma)\) such that

\[
\begin{align*}
\text{min} & \quad J = \beta (t_f) \\
\text{s.t.} & \quad \dot{x} = V \cos \gamma \sin \Psi + W_x, \quad \dot{y} = V \cos \gamma \cos \Psi, \quad \dot{h} = V \sin \gamma \\
& \quad m\dot{V} = -D - mg \sin \gamma - m\dot{W}_x \cos \gamma \sin \Psi \\
& \quad mV\dot{\gamma} = L \cos \sigma - mg \cos \gamma + mW_x \sin \gamma \sin \Psi \\
& \quad mV \cos \gamma \dot{\Psi} = L \sin \sigma - m\dot{W}_x \cos \Psi \\
& \quad W_x = \beta V \sin \gamma, \quad \beta = W_x/h, \quad -2 \leq L/(mg) \leq 5 \\
& \quad (x(0), y(0), h(0)) = (0, 0, 0) \\
& \quad (x(t_f), y(t_f), h(t_f), V(t_f), \gamma(t_f), \Psi(t_f)) = (x(0), y(0), h(0), V(0), \gamma(0), \Psi(0) - 2\pi)
\end{align*}
\]

Here, lift \(L\) and drag \(D\) are expressed as

\[
\begin{align*}
L &= qS C_L \\
D &= qS C_D \\
q &= \rho V^2/2 \\
C_D &= C_{D0} + K C^2_L
\end{align*}
\]

The definitions and values of related variables and parameters involved in Equations (77) and (78) are detailed in reference 21.

In order to find out the optimal control \((C_L, \sigma)\) satisfying Equation (77), the method proposed in this paper is used to solve the problem. The initial mesh point is set as \([-1, 1]\), and the initial number of LGR points is set as 7. The relevant parameters of the algorithm in this paper are set as: \(\epsilon = 10^{-6}, \eta = 1, \sigma = 1.2, \rho = 3\). At the same time, the hp method\(^{23}\) and the ph-\((N_{\min}, N_{\max})\) method\(^{20}\) are used to solve the problem, which is compared with the method in this paper. The simulation results are shown in Figure 7.
Figure 6: Optimization results of Example 2
TABLE 2 Comparison of mesh refinement results of Example 2

| Method     | CPU time (s) | Total number of collocation points | Number of mesh intervals | Number of grid iterations |
|------------|--------------|-----------------------------------|--------------------------|--------------------------|
| Improved hp| 3.99         | 250                               | 39                       | 3                        |
| hp         | 5.33         | 430                               | 70                       | 6                        |
| ph-(3,14)  | 18.35        | 967                               | 69                       | 6                        |
| ph-(5,16)  | 12.66        | 833                               | 52                       | 7                        |
| ph-(6,18)  | 15.33        | 829                               | 46                       | 8                        |

FIGURE 7 Optimization results of Example 3
It can be seen from the simulation results that the method proposed in this paper can effectively solve the optimal control problem of a complex system of the form (77), and the other two methods have basically the same solution results (see Figure 7A,C). However, compared with the other two methods, the method in this paper has obvious advantages in mesh refinement, which requires fewer collocation points and mesh iterations (see Figure 7E,F). This is also reflected in the statistical results in Table 3. As shown in Table 3, when the method proposed in this paper is used to solve the problem, the grid only needs to be iterated 3 times, and the total number of collocation points required is significantly less than the other two methods, which has higher solving efficiency (CPU time is 4.24 s).

### 7 | CONCLUSION

In order to improve the solving efficiency of the optimal problem, this paper proposes an improved adaptive hp mesh refinement method. In this method, the sudden change points of the control curve are accurately located, and then the mesh is refined first near these points, which greatly improves the efficiency of the algorithm. At the same time, a segmentation method based on the cumulative sum of curvature is designed to ensure that more mesh points are allocated at the position with larger curvature of the state curve. This method significantly reduces the number of mesh iterations and shortens the calculation time. Simulation results also show that the number of mesh iterations, the number of required collocation points, and the CPU time of the proposed algorithm are less than other methods.

In fact, more mesh points also need to be configured in the position where the numerical solution approximation error is large to improve the solving precision. Therefore, in the future work, the insertion position of mesh points can be more reasonably determined by comprehensively considering the curvature value of state variables and the numerical solution approximation error, so as to further improve the solving efficiency of the algorithm.

### ACKNOWLEDGMENTS

The authors would like to express their sincere thanks to the editor and anonymous reviewers for their helpful suggestions for improving the technical note.

### FUNDING INFORMATION

This work was supported by the National Natural Science Foundation of China (Grant nos. 61873278 and 61773398).

### CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

### DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in research papers “On the problem of optimal thrust programming for a soft lunar landing” at http://doi.org/10.1109/TAC.1964.1105758,53 “Eigenvector approximate dichotomic basis method for solving hyper-sensitive optimal control problems” at
https://doi.org/10.1002/(SICI)1099-1514(200001/02)21:1<1::AID-OCA646>3.0.CO;2-V,54 and “Optimal patterns of glider dynamic soaring” at https://doi.org/10.1002/oca.739.55

ORCID
Jiong Li  https://orcid.org/0000-0002-1408-4503

REFERENCES
1. Benson DA, Huntington GT, Thorvaldsen TP, Rao AV. Direct trajectory optimization and costate estimation via an orthogonal collocation method. J Guid Control Dyn. 2006;29(6):1435-1440.
2. Huntington GT, Rao AV. Optimal reconfiguration of spacecraft formations using the gauss pseudospectral method. J Guid Control Dyn. 2008;31(3):689-698.
3. Gong Q, Ross IM, Kang W, Fahroo F. Connections between the covector mapping theorem and convergence of pseudospectral methods for optimal control. Comput Optim Appl. 2008;41(3):307-335.
4. Garg D, Patterson M, Hager WW, Rao AV, Benson DA, Huntington GT. A unified framework for the numerical solution of optimal control problems using pseudospectral methods. Automatica. 2010;46(11):1843-1851.
5. Garg D, Patterson MA, Francolin C, et al. Direct trajectory optimization and costate estimation of finite-horizon and infinite-horizon optimal control problems using a Radau pseudospectral method. Comput Optim Appl. 2011;49(2):335-358.
6. Garg D, Hager WW, Rao AV. Pseudospectral methods for solving infinite-horizon optimal control problems. Automatica. 2011;47(4):829-837.
7. Patterson MA, Rao AV. Exploiting sparsity in direct collocation Pseudospectral methods for solving optimal control problems. J Spacecr Rockets. 2012;49(2):364-377.
8. Francolin CC, Benson DA, Hager WW, et al. Costate approximation in optimal control using integral Gaussian quadrature orthogonal collocation methods. Optim Control Appl Methods. 2015;36(4):381-397.
9. Gill PE, Murray W, Saunders MA. SNOPT: an SQP algorithm for large-scale constrained optimization (reprinted from SIAM journal optimization, vol 12, pg 979-1006, 2002). Siam Review. 2005;47(1):99-131.
10. Fahroo F, Ross IM. Advances in Pseudospectral Methods for Optimal Control. Faculty Publications; 2008.
11. Gong Q, Ross IM, Kang W et al., On the pseudospectral covector mapping theorem for nonlinear optimal control. Proceedings of the 45th IEEE Conference on Decision and Control, Vols 1–14; 2006; pp. 2683.
12. Gong Q, Kang W, Ross IM. A pseudospectral method for the optimal control of constrained feedback linearizable systems. IEEE Trans Automat Contr. 2006;51(7):1115-1129.
13. Benson D. A Gauss Pseudospectral Transcription for Optimal Control, PhD Thesis. Department of Aeronautics and Astronautics, Massachusetts Institute of Technology; 2004.
14. Betts JT. Practical Methods for Optimal Control Using Nonlinear Programming. SIAM Press; 2001.
15. Jain S, Tsitouras P. Trajectory optimization using multiresolution techniques. J Guid Control Dyn. 2008;31(5):1424-1436.
16. Zhao Y, Tsitouras P. A density-function based mesh refinement algorithm for solving optimal control problems. Paper presented at: Infotech and Aerospace Conference, AIAA Paper 2009–2019, Seattle, Washington; 2009.
17. Hussaini MY, Zang TA. Spectral methods in fluid-dynamics. Annu Rev Fluid Mech. 1987;19:339-367.
18. Fornberg B. A Practical Guide to Pseudospectral Methods. Cambridge University Press; 1998.
19. Trefethen. Spectral Methods Using MATLAB. SIAM Press; 2000.
20. Patterson MA, Hager WW, Rao AV. A ph mesh refinement method for optimal control. Optim Control Appl Methods. 2015;36(4):398-421.
21. Darby CL, Hager WW, Rao AV. An hp-adaptive pseudospectral method for solving optimal control problems. Optim Control Appl Methods. 2011;32(4):476-502.
22. Liu FJ, Hager WW, Rao AV. An hp mesh refinement method for optimal control using discontinuity detection and mesh size reduction. Paper presented at: 2014 IEEE 53rd Annual Conference on Decision and Control (Cdc); 2014; pp. 5868–5873.
23. Daniel P, Ern A, Smears I, Vohralik M. An adaptive hp-refinement strategy with computable guaranteed bound on the error reduction factor. Comput Math Appl. 2018;76(3):967-983.
24. Koeppen N, Ross IM, Wilcox LC, Proulx RJ. Fast mesh refinement in Pseudospectral optimal control. J Guid Control Dyn. 2019;42(4):711-722.
25. Zhao JS, Li S. Mars atmospheric entry trajectory optimization with maximum parachute deployment altitude using adaptive mesh refinement. Acta Astronaut. 2019;160:401-413.
26. Feng ZW, Zhang QB, Ge JQ, et al. Mesh adaptation method for optimal control with non-smooth control using second-generation wavelet. IEEE Access. 2019;7:135076-135086.
27. Agamawi YM, Rao AV. CGPOPS: a C plus plus software for solving multiple-phase optimal control problems using adaptive Gaussian quadrature collocation and sparse nonlinear programming. ACM Trans Math Softw. 2020;46(3):1-38.
28. Li NB, Lei HM, Shao L, et al. Trajectory optimization based on multi-interval mesh refinement method. Math Probl Eng. 2017; 8521368.
30. Xiao L, Lv L, Liu P, et al. A novel adaptive gauss pseudospectral method for nonlinear optimal control of constrained hypersonic re-entry vehicle problem. *Int J Adapt Control Signal Process*. 2018;32(9):1243-1258.

31. Xia WB, Wang WH, Gao C. Trajectory optimization with obstacles avoidance via strong duality equivalent and hp-pseudospectral sequential convex programming. *Optim Control Appl Methods*. 2022;43(2):566-587.

32. Xu Z, Zeng X, Li J, et al. Optimal motion planning of the space manipulator for minimum reaction torque to satellite. *J Harbin Inst Technol*. 2022;29(2): 81-91.

33. Liu Y, Sun Y, Cui D. Optimum path tracking control for the inverse problem of vehicle handling dynamics based on the hp-adaptive Gaussian pseudospectral method. *Math Probl Eng*. 2021;3:1-15.

34. Yazdani AM, Sammut K, Yakimenko OA, Lammas A. Feasibility analysis of using the hp-adaptive Radau pseudospectral method for minimum-effort collision-free docking operations of AUV. *Rob Auton Syst*. 2020;133(23):103641.

35. Lei HM, Liu T, Li D, et al. Adaptive mesh refinement method for optimal control based on Hermite-Legendre-Gauss-Lobatto direct transcription. *J Vibration and Control*. 2017;19(8):6036-6048.

36. Liu FJ, Hager WW, Rao AV. Adaptive mesh refinement method for optimal control using decay rates of Legendre polynomial coefficients. *IEEE Trans Contr Syst Technol*. 2018;26(4):1475-1483.

37. Souzban M, Fard OS, Borzabadi AH. A rapid-based improvement on some mesh refinement strategies in solving optimal control problems. *IMA J Math Control Inf*. 2020;37(2):351-376.

38. Ejlali N, Hosseini SM. Adaptive control parameterization method by density functions for optimal control problems. *IMA J Math Control Inf*. 2020;37(2):497-512.

39. Zhao JS, Li S. Adaptive mesh refinement method for solving optimal control problems using interpolation error analysis and improved data compression. *J Franklin Inst-Eng Appl Math*. 2020;357(3):1603-1627.

40. Sabermahani S, Ordokhani Y, Yousefi SA. Fractional-order Lagrange polynomials: an application for solving delay fractional optimal control problems. *Trans Inst Meas Control*. 2019;41(11):2997-3009.

41. Sabermahani S, Ordokhani Y, Yousefi S. Fractional-order Lagrange polynomials: an application for solving delay fractional optimal control problems. *Trans Inst Meas Control*. 2019;41(11):2997-3009.

42. Sabermahani S, Ordokhani Y. Fibonacci wavelets and Galerkin method to investigate fractional optimal control problems with bibliometric analysis. *J Vib Control*. 2021;27(15):1778-1792.

43. Rabiei K, Ordokhani Y, Babolian E. Numerical solution of 1D and 2D fractional optimal control of system via Bernoulli polynomials. *Int J Appl Comput Math*. 2021;7(15):1-17. doi:10.1007/s40819-017-0435-0

44. Fujihira H. Necessary and sufficient conditions for bang-bang control. *J Optim Theory Appl*. 1978;25(4):549-554.

45. Lu ZW, Tian GY, Onori S. Multistage time-optimal control for synchronization process in electric-driven mechanical transmission with angle alignment considering torque response process. *J Dyn Syst Meas Control Trans ASME*. 2021;143(4):041006.

46. Han YR, Zhao Y, Wang P. Finite-time rate anti-bump switching control for switched systems. *Appl Math Comput*. 2021;401:126086.

47. Vinter RB. *Optimal Control*. Birkhäuser; 2000.

48. Rao AV, Benson DA, Darby C, et al. Algorithm 902: GPOPS, a MATLAB software for solving multiple-phase optimal control problems using the gauss pseudospectral method. *ACM Trans Math Softw*. 2010;37(2):1-39.

**How to cite this article:** Luo C, Li J, Zhou C, Lei H. An improved adaptive hp mesh refinement method in solving optimal control problems. *Optim Control Appl Meth*. 2023;44(4):1828-1853. doi: 10.1002/oca.2956