

# An Improved Adaptive hp Mesh Refinement Method in Solving Optimal Control Problems

Changxin Luo

Air Force Engineering University

Jiong Li (✉ [graceful001@126.com](mailto:graceful001@126.com))

Air Force Engineering University

Chijun Zhou

Air Force Engineering University

Humin Lei

Air Force Engineering University

---

## Research Article

**Keywords:** optimal control , nonlinear programming problem , pseudospectral method , adaptive hp method , mesh refinement

**Posted Date:** May 25th, 2021

**DOI:** <https://doi.org/10.21203/rs.3.rs-526882/v1>

**License:** © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.  
[Read Full License](#)

---

# An Improved Adaptive hp Mesh Refinement Method in Solving Optimal Control Problems

Changxin Luo · Jiong Li · Chijun Zhou · Humin Lei

Received: date / Accepted: date

**Abstract** The adaptive hp pseudospectral method is an effective choice in solving the optimal control problem. In order to improve the computing efficiency of the adaptive hp method, an improved adaptive hp mesh refinement method is proposed in this paper. There are two main cores in this method. The first is to refine the mesh in advance based on the change rate of the control, so that the algorithm can more efficiently deal with the optimal control problem with discontinuous control. And the second is to based on the accumulation of curvature values to ensure that those positions that need to be refined more can be quickly allocated to more mesh points. These two are combined with each other to improve the solving efficiency of the algorithm proposed in this paper, which shortens the required computing time. In addition, the convergence of the algorithm is proved in this paper. The simulation results show the effectiveness and superiority of the algorithm proposed in this paper.

**Keywords** optimal control · nonlinear programming problem · pseudospectral method · adaptive hp method · mesh refinement

---

C. Luo  
Graduate school, Air Force Engineering University, Xi'an 710051, China

J. Li (Corresponding author)  
Air and Missile Defense College, Air Force Engineering University, Xi'an 710051, China  
E-mail: graceful001@126.com

C. Zhou, H. Lei  
Air and Missile Defense College, Air Force Engineering University, Xi'an 710051, China

## 1 Introduction

In recent years, it has become more and more popular to use direct collocation method to solve complex optimal control problems, among which pseudospectral method [1–8] is developing rapidly. Based on Legendre-Gauss (LG) points [1], Legendre-Gauss-Radau (LGR) points [5,6], or Legendre-Gauss-Lobatto (LGL) points [4], the pseudospectral method mainly uses Lagrangian interpolation polynomials to transform the optimal control problem into nonlinear programming problem (NLP), and then use the NLP solver [9] to solve it.

Many researchers have actively explored the theory and application of the pseudospectral method, and have achieved some considerable results in solving the optimal control problem. Garg et al. [4–6] study the algorithm framework and related theoretical foundation of pseudospectral method based on three types of collocation points of LG, LGR and LGL. In [4], a unified framework of pseudospectral method for three collocation strategies is proposed, and the accuracy and characteristics of the algorithm under each strategy are deeply analyzed. Aiming at the trajectory optimization problem in finite-horizon and infinite-horizon, the solution accuracy of pseudospectral method based on LGR collocation points is deeply studied in [5]. Similarly, based on the two collocation points of LG and LGR, Garg et al. [6] adopt the pseudospectral method to solve the infinite-horizon optimization control problem, focusing on studying the influence of several different time domain transformation methods on the solution results. Relying on the pseudospectral method to transform the continuous optimal control problem into a nonlinear programming problem, Patterson et al. [10] propose a method to calculate the first and second derivatives of the NLP function, which improves the solution efficiency for NLP. For the optimal control problem with terminal constraints, Fahroo et al. [11] reveal the superiority of the pseu-

dospectral method based on Legendre polynomial in dealing with this type of problem. Gong et al. [12–14] use the covector mapping principle to show that the discrete solution obtained by the pseudospectral method can converge to a continuous optimized solution when solving the optimal control problem.

However, when the pseudospectral method is used to solve some complex optimal control problems, it may not be able to obtain a satisfactory solution using a fixed number of collocation points and a single grid interval. In response to this problem, scholars have proposed two strategies. One is the *p* method [15] which improves the accuracy of the solution by increasing the number of collocation points. The other is the *h* method [16–18] to increase the number of mesh intervals by continuously dividing the time domain to ensure the accuracy of the solution. Along these two directions, many researchers have conducted a lot of meaningful research. If there is a smooth analytical solution to the optimal control problem, the *p* method can get good results. Hussaini et al. [19–21] has shown that in this case, by increasing the number of collocations, 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 the solution, 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. Therefore, in this case, *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 [16–18].

Considering that the *h* method and the *p* method have their own advantages, some people have appropriately combined the two to further improve the accuracy of the solution and the efficiency of the calculation [22–25]. Patterson et al. [22] propose a mesh refinement strategy called *ph* method. In essence, it is a way of using *h* method first and then *p* method in the optimization process. In other words, when using the *ph* method for mesh refinement, the order of the interpolation polynomial will first be increased to improve the accuracy of the solution. And when the order reaches the specified maximum limit, the mesh will be segmented instead until a satisfactory solution accuracy is obtained. In [23–25], a mesh refinement method called *hp* method is proposed. Unlike *ph* method, this method is to adaptively decide whether to use *h* method or *p* method to refine the mesh through certain criteria. Darby et al. [23] take the consistency of approximation error as the criterion. If the approximation error of each collocation is relatively consistent, the *p* method is used to refine mesh, otherwise, the *h* method is used. Liu et al. [24,25] take the smoothness of the solu-

tion as the criterion. If the obtained solution is smooth, the *h* method is used for refinement, otherwise, the *p* method is used. The *hp* method has been favored by researchers due to its efficient solving ability and the flexibility of the algorithm. In recent years, many scholars have carried out extensive research on it and greatly developed its related theory and application [26–30]. Li et al. [31,32] apply the adaptive *hp* method to the field of trajectory optimization, which effectively solve this kind of complex nonlinear optimization problem, and obtain satisfactory solution accuracy. By improving the mesh refinement process, some effective variants of the *hp* method are proposed in [33–37].

Although the aforementioned works [23–37] on the adaptive *hp* method has made great progress, there still are several issues that need further consideration.

1. When there are sudden changes in the optimal control curve (such as bang-bang control [38,39], switch control [40]), 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 grid 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 [46], 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 [23–26]) 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

[46] 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. In Section 2, the optimal control problem to be solved in this paper is described and transformed into a nonlinear programming problem based on LGR collocation method. Section 3 is the description of the improved adaptive hp mesh refinement method, and the core algorithm of this paper is introduced in detail. In Section 4, we prove the convergence of the proposed algorithm. The simulation example in the Section 5 verifies the effectiveness and superiority of the method in this paper. In Section 6, the work of this paper is summarized.

## 2 Preliminaries

### 2.1 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  $(\mathbf{x}, \mathbf{u}) \in \mathbf{R}^{n_x} \times \mathbf{R}^{n_u}$  such that

$$(B) \begin{cases} \min J = E(\mathbf{x}_0, \mathbf{x}_f, t_0, t_f) + \int_{t_0}^{t_f} F(\mathbf{x}(t), \mathbf{u}(t)) dt \\ s.t. \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{e}(\mathbf{x}_0, \mathbf{x}_f, t_0, t_f) = \mathbf{0} \\ \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t)) \leq \mathbf{0} \end{cases} \quad (1)$$

Where,  $\mathbf{x} \in \mathbf{R}^{n_x}$  is the state variable,  $\mathbf{u} \in \mathbf{R}^{n_u}$  is the control variable, and  $\mathbf{x}$  and  $\mathbf{u}$  are defined as row vectors.  $t_0$  and  $t_f$  represent initial time and terminal time, respectively, and  $\mathbf{x}_0 = \mathbf{x}(t_0)$ ,  $\mathbf{x}_f = \mathbf{x}(t_f)$ .  $E(\bullet)$  and  $F(\bullet)$  are scalar functions, which represent the terminal type and integral type performance indicators, respectively.  $\mathbf{f}(\bullet) \in \mathbf{R}^{n_f}$  represents dynamic constraints,  $\mathbf{e}(\bullet) \in \mathbf{R}^{n_e}$  represents boundary conditions,  $\mathbf{h}(\bullet) \in \mathbf{R}^{n_h}$  represents inequality path constraints. Define  $\mathbf{f}$ ,  $\mathbf{e}$  and  $\mathbf{h}$  as row vectors. All these nonlinear functions ( $E, F, \mathbf{f}, \mathbf{e}, \mathbf{h}$ ) are continuously differentiable in their domains, and their gradients are Lipschitz continuous.

According to Pontryagin minimum principle [41], we can reduce the first-order necessary condition of problem B to the following boundary-value problem [42, 43].

$$(B^\lambda) \begin{cases} \text{find } [\mathbf{x}(\bullet), \mathbf{u}(\bullet), \boldsymbol{\lambda}(\bullet), \boldsymbol{\mu}(\bullet), \mathbf{v}] \\ \text{such that } \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{e}(\mathbf{x}_0, \mathbf{x}_f, t_0, t_f) = \mathbf{0} \\ \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t)) \leq \mathbf{0} \\ \dot{\boldsymbol{\lambda}}(t) = -\partial \bar{H}(t) / \partial \mathbf{x} \\ \partial \bar{H} / \partial \mathbf{u} = \mathbf{0} \\ \{\boldsymbol{\lambda}(t_0), \boldsymbol{\lambda}(t_f)\} = \{-\partial \bar{E} / \partial \mathbf{x}_0, \partial \bar{E} / \partial \mathbf{x}_f\} \\ \{H(t_0), H(t_f)\} = \{\partial \bar{E} / \partial t_0, -\partial \bar{E} / \partial t_f\} \\ \boldsymbol{\mu}(t) \geq \mathbf{0}, \boldsymbol{\mu}(t) \cdot \mathbf{h}^T(\mathbf{x}(t), \mathbf{u}(t)) = 0 \end{cases} \quad (2)$$

Where  $\bar{H}(t)$  represents the Lagrangian of the Hamiltonian.

$$\bar{H}(\boldsymbol{\lambda}, \boldsymbol{\mu}, \mathbf{x}, \mathbf{u}) := H(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{u}) + \boldsymbol{\mu} \cdot \mathbf{h}^T(\mathbf{x}, \mathbf{u}) \quad (3)$$

$H$  is the Hamiltonian function.

$$H(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{u}) := F(\mathbf{x}, \mathbf{u}) + \boldsymbol{\lambda} \cdot \mathbf{f}^T(\mathbf{x}, \mathbf{u}) \quad (4)$$

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

$$\bar{E}(\mathbf{v}, \mathbf{x}_0, \mathbf{x}_f, t_0, t_f) := E(\mathbf{x}_0, \mathbf{x}_f, t_0, t_f) + \mathbf{v} \mathbf{e}^T(\mathbf{x}_0, \mathbf{x}_f, t_0, t_f) \quad (5)$$

In addition, the Lagrangian multipliers  $\boldsymbol{\mu}$ ,  $\boldsymbol{\lambda}$ , and  $\mathbf{v}$  in Eqs. (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  $[T_0, T_1, \dots, T_K]$  on the entire time interval  $[t_0, t_f]$ , and divide them into  $K$  mesh intervals  $S_k = [T_{k-1}, T_k]$ ,  $k = 1, 2, \dots, K$ . Where  $T_0 = t_0$ ,  $T_K = t_f$ ,  $\bigcup_{k=1}^K S_k = [t_0, t_f]$ . Let  $\mathbf{x}^{(k)}(t)$  and  $\mathbf{u}^{(k)}(t)$  denote the state and the control on the mesh interval  $S_k$ , respectively. Therefore, problem B (Eq. (1)) can be equivalent to

$$(B) \begin{cases} \min J = E(\mathbf{x}^{(1)}(T_0), \mathbf{x}^{(K)}(T_K)) \\ + \sum_{k=1}^K \int_{T_{k-1}}^{T_k} F(\mathbf{x}^{(k)}(t), \mathbf{u}^{(k)}(t)) dt \\ s.t. \frac{d\mathbf{x}^{(k)}(t)}{dt} = \mathbf{f}(\mathbf{x}^{(k)}(t), \mathbf{u}^{(k)}(t)) \quad k = 1, 2, \dots, K \\ \mathbf{e}(\mathbf{x}^{(1)}(T_0), \mathbf{x}^{(K)}(T_K)) = \mathbf{0} \\ \mathbf{h}(\mathbf{x}^{(k)}(t), \mathbf{u}^{(k)}(t)) \leq \mathbf{0} \quad k = 1, 2, \dots, K \end{cases} \quad (6)$$

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

$$\mathbf{x}^{(k)}(T_k^-) = \mathbf{x}^{(k+1)}(T_k^+), \quad k = 1, 2, \dots, K-1 \quad (7)$$

## 2.2 Legendre-Gauss-Radau Collocation

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

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

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

$$(B) \begin{cases} \min J = E(\mathbf{x}^{(1)}(-1), \mathbf{x}^{(K)}(+1)) \\ \quad + \sum_{k=1}^K \frac{h_k}{2} \int_{-1}^{+1} F(\mathbf{x}^{(k)}(\tau), \mathbf{u}^{(k)}(\tau)) d\tau \\ \text{s.t. } \frac{d\mathbf{x}^{(k)}(\tau)}{d\tau} = \frac{h_k}{2} \mathbf{f}(\mathbf{x}^{(k)}(\tau), \mathbf{u}^{(k)}(\tau)) \quad k = 1, 2, \dots, K \\ \mathbf{e}(\mathbf{x}^{(1)}(-1), \mathbf{x}^{(K)}(+1)) = \mathbf{0} \\ \mathbf{h}(\mathbf{x}^{(k)}(\tau), \mathbf{u}^{(k)}(\tau)) \leq \mathbf{0} \quad k = 1, 2, \dots, K \\ \mathbf{x}^{(k)}(+1) = \mathbf{x}^{(k+1)}(-1), \quad k = 1, 2, \dots, K-1 \end{cases} \quad (9)$$

Then, select  $N_k$  LGR collocation points  $\tau_1^{(k)}, \dots, \tau_{N_k}^{(k)} \in [-1, 1]$  [44] on each mesh interval  $S_k$  ( $k = 1, \dots, K$ ), where  $\tau_1^{(k)}, \dots, \tau_{N_k}^{(k)}$  ( $\tau_1^{(k)} = -1$ ) 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 Eq. (9), we use LGR collocation points  $\tau_1^{(k)}, \dots, \tau_{N_k}^{(k)}$  and noncollocated point  $\tau_{N_k+1}^{(k)} = +1$  as sampling points on each mesh interval  $S_k$  to achieve Lagrange interpolation on  $\mathbf{x}^{(k)}(\tau)$ . The interpolation polynomial based on these  $N_k + 1$  points is expressed as follows

$$l_j^{(k)}(\tau) = \prod_{\substack{i=1 \\ i \neq j}}^{N_k+1} \frac{\tau - \tau_i^{(k)}}{\tau_j^{(k)} - \tau_i^{(k)}}, \quad j = 1, 2, \dots, N_k + 1 \quad (10)$$

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

$$\mathbf{x}^{(k)}(\tau) \approx \mathbf{X}^{(k)}(\tau) = \sum_{j=1}^{N_k+1} \mathbf{X}_j^{(k)} l_j^{(k)}(\tau) \quad (11)$$

Here,  $\mathbf{X}_j^{(k)}$ ,  $j = 1, 2, \dots, N_k + 1$  is an approximation of  $\mathbf{x}^{(k)}(\tau_j^{(k)})$ . Taking the derivative of Eq. (11) at the collocation points  $\tau_1^{(k)}, \dots, \tau_{N_k}^{(k)}$ , we have

$$\dot{\mathbf{x}}^{(k)}(\tau_i^{(k)}) = \sum_{j=1}^{N_k+1} \mathbf{X}_j^{(k)} l_j^{(k)}(\tau_i^{(k)}) = \sum_{j=1}^{N_k+1} \mathbf{X}_j^{(k)} D_{ij}^{(k)}, \quad i = 1, \dots, N_k \quad (12)$$

Where,  $D_{ij}^{(k)} = l_j^{(k)}(\tau_i^{(k)})$ . Let  $\mathbf{U}_j^{(k)}$ ,  $j = 1, \dots, N_k$  denote the approximation of the control  $\mathbf{u}^{(k)}(\tau_j^{(k)})$ . Then, by interpolating the function  $F(\mathbf{x}^{(k)}(\tau), \mathbf{u}^{(k)}(\tau))$ , its integral can be approximated as

$$\int_{-1}^{+1} F(\mathbf{x}^{(k)}(\tau), \mathbf{u}^{(k)}(\tau)) d\tau = \sum_{j=1}^{N_k} F(\mathbf{X}_j^{(k)}, \mathbf{U}_j^{(k)}) \omega_j^{(k)} \quad (13)$$

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

$$\omega_j^{(k)} = \int_{-1}^{+1} l_j^{(k)}(\tau) d\tau \quad (14)$$

Combining Eqs. (10)~(14), we can transform the optimal control problem (Eq. (9)) into an NLP, which is given as problem  $B^N$

$$(B^N) \begin{cases} \min J = E(\mathbf{X}_1^{(1)}, \mathbf{X}_{N_K+1}^{(K)}) + \sum_{k=1}^K \sum_{j=1}^{N_k} \frac{h_k}{2} \omega_j^{(k)} F(\mathbf{X}_j^{(k)}, \mathbf{U}_j^{(k)}) \\ \text{s.t. } \sum_{j=1}^{N_k+1} \mathbf{X}_j^{(k)} D_{ij}^{(k)} = \frac{h_k}{2} \mathbf{f}(\mathbf{X}_i^{(k)}, \mathbf{U}_i^{(k)}) \quad i = 1, \dots, N_k \\ \mathbf{e}(\mathbf{X}_1^{(1)}, \mathbf{X}_{N_K+1}^{(K)}) = \mathbf{0} \\ \mathbf{h}(\mathbf{X}_i^{(k)}, \mathbf{U}_i^{(k)}) \leq \mathbf{0} \quad i = 1, 2, \dots, N_k \\ \mathbf{X}_{N_k+1}^{(k)} = \mathbf{X}_1^{(k+1)}, \quad k = 1, 2, \dots, K-1 \end{cases} \quad (15)$$

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  $[T_0, T_1, \dots, T_K]$  (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.

### 3 Proposed Algorithm Framework

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 [22–26], 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  $\dot{U}_i^{(k)}$  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 3.1 is the evaluation of the solution error. In Section 3.2~3.6, we give a detailed description of the core algorithm of mesh refinement proposed in this paper.

#### 3.1 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 Eq. (11) can be within a tolerance range  $\varepsilon$ . Since we cannot know the analytical solution  $x^{(k)}(\tau)$  of problem  $B$  in advance, here 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  $S_k$  ( $k = 1, 2, \dots, K$ ), denote the new  $\Gamma_k$  collocation points as  $(\hat{\tau}_1^{(k)}, \dots, \hat{\tau}_{\Gamma_k}^{(k)})$ , 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  $(X_j^{(k)}, U_j^{(k)})$  and the corresponding sampling points  $(\tau_1^{(k)}, \dots, \tau_{N_k+1}^{(k)})$ , the state  $X^{(k)}(\hat{\tau}_i^{(k)})$  at  $(\hat{\tau}_1^{(k)}, \dots, \hat{\tau}_i^{(k)}, \dots, \hat{\tau}_{\Gamma_k+1}^{(k)})$  and the control  $U^{(k)}(\hat{\tau}_i^{(k)})$  at  $(\hat{\tau}_1^{(k)}, \dots, \hat{\tau}_i^{(k)}, \dots, \hat{\tau}_{\Gamma_k}^{(k)})$  are obtained by Lagrange interpolation. Therefore,  $X^{(k)}(\hat{\tau}_i^{(k)})$  can be approximately expressed as

$$\begin{cases} X^{(k)}(\hat{\tau}_i^{(k)}) = \sum_{j=1}^{N_k+1} X_j^{(k)} l_j^{(k)}(\hat{\tau}_i^{(k)}), & i = 1, \dots, \Gamma_k + 1 \\ l_j^{(k)}(\tau) = \prod_{\substack{i=1 \\ i \neq j}}^{N_k+1} \frac{\tau - \tau_i^{(k)}}{\tau_j^{(k)} - \tau_i^{(k)}}, & j = 1, 2, \dots, N_k + 1 \end{cases} \quad (16)$$

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

$$\begin{cases} U^{(k)}(\hat{\tau}_i^{(k)}) = \sum_{j=1}^{N_k} U_j^{(k)} l_j^{(k)}(\hat{\tau}_i^{(k)}), & i = 1, \dots, \Gamma_k \\ l_j^{(k)}(\tau) = \prod_{\substack{i=1 \\ i \neq j}}^{N_k} \frac{\tau - \tau_i^{(k)}}{\tau_j^{(k)} - \tau_i^{(k)}}, & j = 1, 2, \dots, N_k \end{cases} \quad (17)$$

Finally, combining the  $X^{(k)}(\hat{\tau}_i^{(k)})$  and  $U^{(k)}(\hat{\tau}_i^{(k)})$  obtained above, integrate again to obtain the state  $\hat{X}^{(k)}(\hat{\tau}_i^{(k)})$  at  $(\hat{\tau}_1^{(k)}, \dots, \hat{\tau}_i^{(k)}, \dots, \hat{\tau}_{\Gamma_k+1}^{(k)})$  using the integral form of the discrete state equation.

For the new LGR collocation points  $(\hat{\tau}_1^{(k)}, \dots, \hat{\tau}_{\Gamma_k}^{(k)})$ , construct a  $\Gamma_k \times (\Gamma_k + 1)$  differentiation matrix  $\hat{D}^{(k)}$ , and the element at position  $(i, j)$  is  $\hat{D}_{ij}^{(k)} = d\hat{l}_j^{(k)}(\hat{\tau}_i^{(k)})/d\tau$ , where

$$\hat{l}_j^{(k)}(\tau) = \prod_{\substack{i=1 \\ i \neq j}}^{\Gamma_k+1} \frac{\tau - \hat{\tau}_i^{(k)}}{\hat{\tau}_j^{(k)} - \hat{\tau}_i^{(k)}}, \quad j = 1, 2, \dots, \Gamma_k + 1 \quad (18)$$

Similar to the derivation of the discrete state equation in Eq. (15), the discrete state equation at collocation points  $(\hat{\tau}_1^{(k)}, \dots, \hat{\tau}_{\Gamma_k}^{(k)})$  can be expressed as follows

$$\sum_{j=1}^{\Gamma_k+1} \hat{D}_{ij}^{(k)} X^{(k)}(\hat{\tau}_j^{(k)}) = \frac{h_k}{2} f(X^{(k)}(\hat{\tau}_i^{(k)}), U^{(k)}(\hat{\tau}_i^{(k)})) \quad (19)$$

$i = 1, 2, \dots, \Gamma_k \quad k = 1, 2, \dots, K$

At the same time, an  $\Gamma_k \times \Gamma_k$  integration matrix  $\hat{I}^{(k)} = [\hat{D}_2^{(k)}, \dots, \hat{D}_{\Gamma_k+1}^{(k)}]^{-1}$  is constructed, where  $\hat{D}_i^{(k)}$  is the  $i$ th column element of the differentiation matrix  $\hat{D}^{(k)}$ . Therefore, combined with Eq. (19), we can obtain the integral form of the discrete state equation as follows

$$\begin{aligned} \hat{X}^{(k)}(\hat{\tau}_{i+1}^{(k)}) &= X^{(k)}(\hat{\tau}_1^{(k)}) \\ &+ \frac{h_k}{2} \sum_{j=1}^{\Gamma_k} \hat{l}_{ij}^{(k)} f(X^{(k)}(\hat{\tau}_i^{(k)}), U^{(k)}(\hat{\tau}_i^{(k)})), \quad i = 1, \dots, \Gamma_k \end{aligned} \quad (20)$$

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

So far, we can use  $|X^{(k)}(\hat{\tau}_i^{(k)}) - \hat{X}^{(k)}(\hat{\tau}_i^{(k)})|$  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

$$\begin{cases} e_l^{(k)}(\hat{\tau}_i^{(k)}) = |X_l^{(k)}(\hat{\tau}_i^{(k)}) - \hat{X}_l^{(k)}(\hat{\tau}_i^{(k)})| \\ e_{\max}^{(k)} = \max_{\substack{l=1, \dots, n_x \\ i=1, \dots, I_k+1}} e_l^{(k)}(\hat{\tau}_i^{(k)}) \end{cases} \quad (21)$$

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

On the mesh interval  $S_k = [T_{k-1}, T_k]$ , we use  $e_{\max}^{(k)}$  calculated in Eq. (21) to represent the error of numerical solution  $(X_j^{(k)}, U_j^{(k)})$ . Set  $\varepsilon$  is the accuracy tolerance range. If  $e_{\max}^{(k)} \leq \varepsilon$  holds for all mesh interval  $S_k$ ,  $k = 1, \dots, K$ , 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.

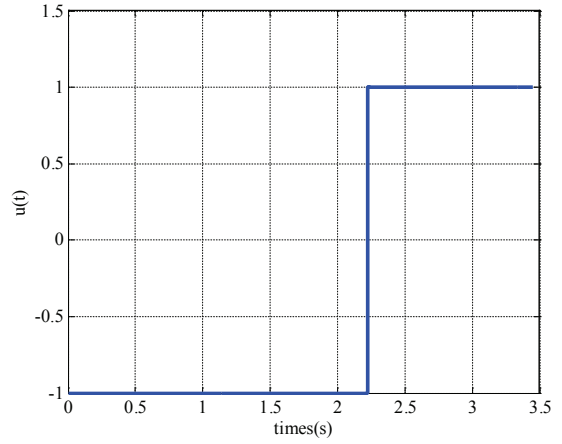
### 3.2 The Mesh Refinement in Advance Based on Control Sudden Changes

For many optimal control problems, there are some sudden changes on the control curve, such as bang-bang control [38,39], switching control [40] and so on. When hp pseudospectral method is used to solve problem  $B^N$ , the sudden changes of control curve will greatly increase the number of iterations and affect the efficiency of solution. For example, consider the time optimal control problem of the following second-order integrator model

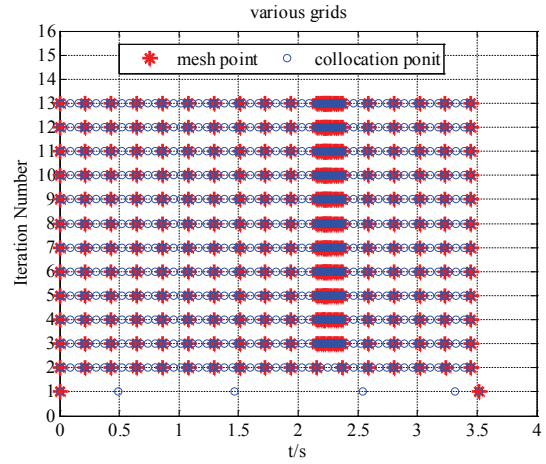
$$\begin{cases} \min J = \int_0^{t_f} dt \\ s.t. \dot{x}_1(t) = x_2(t), \dot{x}_2(t) = u(t) \\ x_1(0) = 1, x_1(t_f) = 0 \\ x_2(0) = 1, x_2(t_f) = 0 \\ -1 \leq u(t) \leq 1 \end{cases} \quad (22)$$

In order to find out the optimal control  $u^*(t)$  to minimize the cost function  $J$ , the adaptive hp method in [23] is used to solve Eq. (22). The results are as in Fig.1.

According to the optimal control theory, this example is bang-bang control, and its analytical solution is shown in Eq.(23).



(a) control vs time



(b) approximation points of various grids

Fig. 1 Optimization results of system (22)

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

By comparing Eq. (23) with Fig. 1(a), when the adaptive hp method [23] 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 Fig. 1 (b)).

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.

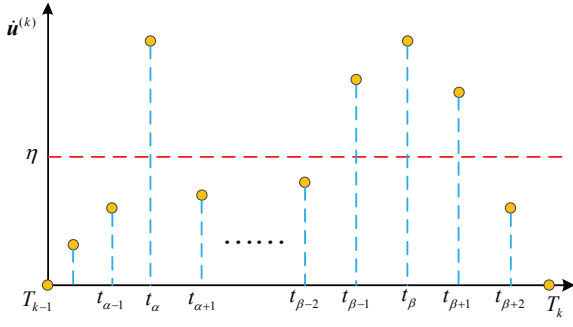


Fig. 2 The derivative of control.

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

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

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

$$\dot{u}^{(k)}(t_i) = \frac{2}{h_k} \frac{u^{(k)}(\tau_{i+1}) - u^{(k)}(\tau_i)}{\tau_{i+1} - \tau_i} = \frac{2}{h_k} \frac{U_{i+1}^{(k)} - U_i^{(k)}}{\tau_{i+1} - \tau_i} \quad (25)$$

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 Fig.2.

In the Fig. 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 Fig. 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 Fig. 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  $[T'_0, T'_1, \dots, T'_{K'}]$ , and the new mesh intervals are denoted as  $S'_k = [T'_{k-1}, T'_k]$ ,  $k = 1, 2, \dots, K'$ .

### 3.3 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 [22–25]. 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  $\Omega_{t_k}$ , and the state at each collocation point is expressed as  $\Omega_{x(t_k)}$ . Based on  $\Omega_{t_k}$  and  $\Omega_{x(t_k)}$ ,  $N$  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  $(\chi_i^{(k)}, Y_i^{(k)})$ ,  $i = 0, 1, \dots, N-1$ , 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.

$$\kappa_i^{(k)} = \frac{|\ddot{Y}_i^{(k)}|}{\left|1 + (\dot{Y}_i^{(k)})^2\right|^{\frac{3}{2}}} \quad i = 0, 1, \dots, N-1 \quad (26)$$

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

$$Cri^{(k)} = \frac{\max_{i=0,1,\dots,N-1} \kappa_i^{(k)}}{\sum_{i=0}^{N-1} \kappa_i^{(k)} / N} \quad (27)$$

If the curvature threshold parameter is set to  $\rho$  and  $Cri^{(k)} > \rho$  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  $Cri^{(k)} \leq \rho$ , it is necessary to refine the mesh by adding collocation points.

### 3.4 Increasing Collocation Points based on Maximum Approximation Error

If it is determined in Section 3.3 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  $e_{\max}^{(k)}$  in Eq. (21) in the following way

$$N'_k = N_k + \left\lceil \log_{10} \left( \frac{e_{\max}^{(k)}}{\varepsilon} \right) \right\rceil \quad (28)$$

Where,  $N_k$  is the original allocation number of interval  $S'_k$ ,  $N'_k$  is the increased allocation number, and  $\varepsilon$  is the set accuracy tolerance range.  $\lceil \bullet \rceil$  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.



### 3.5 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

$$M^{(k)} = \max \left( 2, \left\lceil \sigma \log_{10} \left( \frac{e_{\max}^{(k)}}{\varepsilon} \right) \right\rceil \right) \quad (29)$$

Where  $\max(\circ, *)$  is the maximum of  $\circ$  and  $*$ .  $\sigma$  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 Eq. (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  $\kappa_i^{(k)}$  at each interpolation point  $(\chi_i^{(k)}, Y_i^{(k)})$  in Eq. (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 Fig. 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  $\chi_i^{(k)}$  in the interval  $S'_k$ , and record them as  $m_i^{(k)}, i = 1, \dots, M^{(k)} - 1$ , so that the sum of the curvatures at the interpolation points in each sub interval

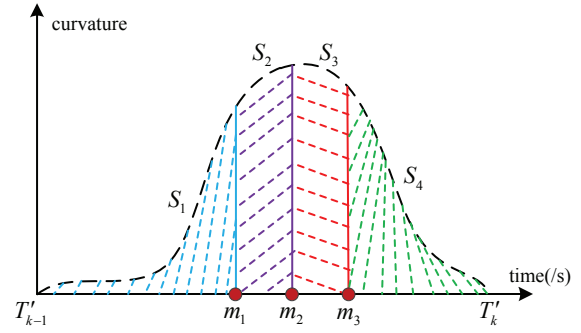


Fig. 3 Mesh interval segmentation.

$[m_{i-1}^{(k)}, m_i^{(k)}]$  is approximately equal. According to this rule, the algorithm for selecting new mesh points is as follows

$$m_i^{(k)} = \left\{ \chi_j^{(k)} \mid \sum_{t=0}^j Y_t^{(k)} \geq i\bar{S}, \sum_{t=0}^{j-1} Y_t^{(k)} < i\bar{S} \right\} \quad (30)$$

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

### 3.6 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

- Step 1. Select the initial mesh points  $[T_0, T_1, \dots, T_K]$ , and set the initial collocation points  $(\tau_1^{(k)}, \dots, \tau_{N_k}^{(k)})$  in each mesh interval  $S_k = [T_{k-1}, T_k]$ .
- 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  $(X_j^{(k)}, U_j^{(k)})$ .
- Step 3. Use the method in Section 3.1 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.
- Step 4. Complete the iteration of the solution, and take the current numerical solution as the final output.
- 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

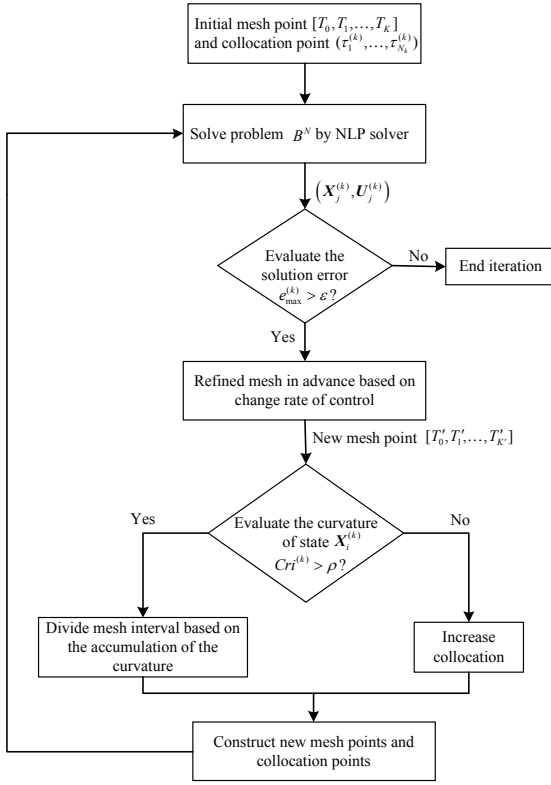


Fig. 4 Algorithm procedure.

original mesh points into  $[T'_0, T'_1, \dots, T'_{K'}]$  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  $C r_i^{(k)}$  in Eq. (27) is greater than the threshold  $\rho$ , go to step iii. Otherwise, go to step iv.
- iii. According to the method in Section E, the mesh interval  $S'_k$  is segmented based on curvature accumulation.
- iv. Based on the maximum approximation error  $e_{\max}^{(k)}$  in Eq. (21), as described in Section 3.4, add a certain number of collocation points in the mesh interval  $S'_k$ .

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 Fig. 4.

#### 4 Theoretical Convergence Results

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

**Assumption 1** For  $l \geq 3$ , in the domain  $C^{l+1}(\mathbf{R}^{n_x}) \times L^\infty(\mathbf{R}^{n_u}) \times C^{l+1}(\mathbf{R}^{n_f}) \times C^{l+1}(\mathbf{R}^{n_h}) \times \mathbf{R}^{n_e}$ , there is an optimal solu-

tion  $(\mathbf{x}^*, \mathbf{u}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*, \mathbf{v}^*)$  for problem  $B$  (Eq. (1)) so that the Pontryagin minimum condition (Eq. (2)) holds. At the same time, there is  $\vartheta > 0$  and open set  $\Omega \subset \mathbf{R}^{n_x} \times \mathbf{R}^{n_u}$  such that  $B_\vartheta(\mathbf{x}^*(t), \mathbf{u}^*(t)) \subset \Omega$  holds for any  $t \in [t_0, t_f]$ , and the first two derivatives of functions  $F(\bullet)$ ,  $f(\bullet)$  and  $h(\bullet)$  are Lipschitz continuous in  $\Omega$ , and the first two derivatives of  $E(\bullet)$  and  $e(\bullet)$  are Lipschitz continuous in  $B_\vartheta(\mathbf{x}^*(t_f))$ .

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

$$\mathbf{A} = \nabla_{xx}E(\mathbf{x}^*(t_f)), \mathbf{B} = \nabla_{ux}e(\mathbf{x}^*(t_f)), \mathbf{C} = \begin{pmatrix} \mathbf{Q}(t) & \mathbf{S}(t) \\ \mathbf{S}^T(t) & \mathbf{R}(t) \end{pmatrix}$$

Where  $\mathbf{Q}(t) = \nabla_{xx}\bar{H}$ ,  $\mathbf{S}(t) = \nabla_{ux}\bar{H}$ ,  $\mathbf{R}(t) = \nabla_{uu}\bar{H}$ . Assume that the smallest eigenvalues of matrices  $\mathbf{A}$ ,  $\mathbf{B}$ , and  $\mathbf{C}$  are strictly greater than zero.

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

Next, analyze the convergence of the improved adaptive hp algorithm proposed in this paper, i.e. to prove that for the NLP  $B^N$  (Eq. (15)) there is a minimum point  $(\mathbf{X}_i^{(k)}, \mathbf{U}_i^{(k)})$  and Lagrangian multiplier vectors  $\boldsymbol{\lambda}_i^{(k)}$ ,  $\boldsymbol{\mu}_i^{(k)}$  and  $\mathbf{v}^{(k)}$  such that  $(\mathbf{X}_i^{(k)}, \mathbf{U}_i^{(k)}, \boldsymbol{\lambda}_i^{(k)}, \boldsymbol{\mu}_i^{(k)}, \mathbf{v}^{(k)})$  converges to the optimal solution  $(\mathbf{X}_i^{*(k)}, \mathbf{U}_i^{*(k)}, \boldsymbol{\lambda}_i^{*(k)}, \boldsymbol{\mu}_i^{*(k)}, \mathbf{v}^{*(k)})$  of problem  $B$ , where  $k = 1, \dots, K$ ,  $i = 1, \dots, N_k$ ,  $\mathbf{X}_i^{*(k)} = \mathbf{x}^{*(k)}(\tau_i)$ ,  $\mathbf{U}_i^{*(k)} = \mathbf{u}^{*(k)}(\tau_i)$ ,  $\boldsymbol{\lambda}_i^{*(k)} = \boldsymbol{\lambda}^{*(k)}(\tau_i)$ ,  $\boldsymbol{\mu}_i^{*(k)} = \boldsymbol{\mu}^{*(k)}(\tau_i)$ .

**Theorem 1** If Assumption 1 ~ Assumption 3 hold, then for problem  $B^N$  (Eq. (15)), there is a minimum point  $(\mathbf{X}_i^{(k)}, \mathbf{U}_i^{(k)})$  and Lagrangian multiplier vectors  $\boldsymbol{\lambda}_i^{(k)}$ ,  $\boldsymbol{\mu}_i^{(k)}$  and  $\mathbf{v}^{(k)}$  satisfy

$$\max \left\{ \left\| \mathbf{X}^{N_k} - \mathbf{X}^{*(k)} \right\|_\infty, \left\| \mathbf{U}^{N_k} - \mathbf{U}^{*(k)} \right\|_\infty, \left\| \boldsymbol{\lambda}^{N_k} - \boldsymbol{\lambda}^{*(k)} \right\|_\infty, \left\| \boldsymbol{\mu}^{N_k} - \boldsymbol{\mu}^{*(k)} \right\|_\infty, \left\| \mathbf{v}^{(k)} - \mathbf{v}^{*(k)} \right\|_\infty \right\} \leq \frac{C_k h_k^{l_k+1}}{N_k^{l_k - \frac{5}{2}}} \quad (31)$$

Where  $\mathbf{X}^{N_k} = (\mathbf{X}_1^{(k)}, \dots, \mathbf{X}_{N_k}^{(k)})^T$ ,  $\mathbf{X}^{*(k)} = (\mathbf{X}_1^{*(k)}, \dots, \mathbf{X}_{N_k}^{*(k)})^T$ ,  $\mathbf{U}^{N_k} = (\mathbf{U}_1^{(k)}, \dots, \mathbf{U}_{N_k}^{(k)})^T$ ,  $\mathbf{U}^{*(k)} = (\mathbf{U}_1^{*(k)}, \dots, \mathbf{U}_{N_k}^{*(k)})^T$ ,  $\boldsymbol{\lambda}^{N_k} = (\boldsymbol{\lambda}_1^{(k)}, \dots, \boldsymbol{\lambda}_{N_k}^{(k)})^T$ ,  $\boldsymbol{\lambda}^{*(k)} = (\boldsymbol{\lambda}_1^{*(k)}, \dots, \boldsymbol{\lambda}_{N_k}^{*(k)})^T$ ,  $\boldsymbol{\mu}^{N_k} = (\boldsymbol{\mu}_1^{(k)}, \dots, \boldsymbol{\mu}_{N_k}^{(k)})^T$ ,  $\boldsymbol{\mu}^{*(k)} = (\boldsymbol{\mu}_1^{*(k)}, \dots, \boldsymbol{\mu}_{N_k}^{*(k)})^T$ . In addition,  $k = 1, \dots, K$ .  $N_k$  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  $B^N$  (Eq. (15)).

Introduce Lagrangian multiplier vectors  $\mathbf{\Lambda}_i^{(k)} \in \mathbf{R}^{n_f}, \mathbf{m}_i^{(k)} \in \mathbf{R}^{n_h}$  and  $\boldsymbol{\xi}^{(k)} \in \mathbf{R}^{n_e}$  (all three are assumed to be row vectors), and construct the following Lagrangian function

$$\begin{aligned} L(\mathbf{X}_i^{(k)}, \mathbf{U}_i^{(k)}, \mathbf{\Lambda}_i^{(k)}, \mathbf{m}_i^{(k)}, \boldsymbol{\xi}^{(k)}) \\ = E(\mathbf{X}_1^{(1)}, \mathbf{X}_{N_K+1}^{(K)}) + \sum_{k=1}^K \sum_{j=1}^{N_k} \frac{h_k}{2} \omega_j^{(k)} F(\mathbf{X}_j^{(k)}, \mathbf{U}_j^{(k)}) \\ + \sum_{k=1}^K \sum_{j=1}^{N_k} \mathbf{\Lambda}_i^{(k)} \cdot \left( \frac{h_k}{2} \mathbf{f}(\mathbf{X}_i^{(k)}, \mathbf{U}_i^{(k)}) - \sum_{j=1}^{N_k+1} \mathbf{X}_j^{(k)} D_{ij}^{(k)} \right)^T \\ + \boldsymbol{\xi}^{(k)} \cdot \mathbf{e}^T (\mathbf{X}_1^{(1)}, \mathbf{X}_{N_K+1}^{(K)}) + \sum_{k=1}^K \sum_{j=1}^{N_k} \mathbf{m}_i^{(k)} \cdot \mathbf{h}^T (\mathbf{X}_i^{(k)}, \mathbf{U}_i^{(k)}) \end{aligned} \quad (32)$$

Let  $D_i^{(k)}$  denote the element in column  $i$  of differentiation matrix  $\mathbf{D}^{(k)} = [D_{ij}^{(k)}]$ . At the same time  $\mathbf{\Lambda}^{(k)} = (\mathbf{\Lambda}_1^{(k)}, \dots, \mathbf{\Lambda}_{N_k}^{(k)})^T$ . Calculate the gradient of Lagrangian function  $L$  in Eq. (32) to  $\mathbf{X}_{N_K+1}^{(K)}, \mathbf{X}_1^{(1)}, \mathbf{X}_j^{(k)}$  and  $\mathbf{U}_j^{(k)}$  respectively. The first-order necessary conditions for the minimum point are as follows

$$\begin{aligned} \nabla_{\mathbf{X}_{N_K+1}^{(K)}} \left( E(\mathbf{X}_1^{(1)}, \mathbf{X}_{N_K+1}^{(K)}) + \boldsymbol{\xi}^{(k)} \cdot \mathbf{e}^T (\mathbf{X}_1^{(1)}, \mathbf{X}_{N_K+1}^{(K)}) \right) \\ - \left( \mathbf{D}_{N_K+1}^{(K)} \right)^T \cdot \mathbf{\Lambda}^{(K)} = 0 \end{aligned} \quad (33)$$

$$\begin{aligned} \nabla_{\mathbf{X}_1^{(1)}} \left( E(\mathbf{X}_1^{(1)}, \mathbf{X}_{N_K+1}^{(K)}) + \boldsymbol{\xi}^{(k)} \cdot \mathbf{e}^T (\mathbf{X}_1^{(1)}, \mathbf{X}_{N_K+1}^{(K)}) \right) \\ + \frac{h_1}{2} \omega_1^{(1)} \nabla_{\mathbf{X}_1^{(1)}} F(\mathbf{X}_1^{(1)}, \mathbf{U}_1^{(1)}) + \frac{h_1}{2} \mathbf{\Lambda}_1^{(1)} \nabla_{\mathbf{X}_1^{(1)}} \mathbf{f}^T(\mathbf{X}_1^{(1)}, \mathbf{U}_1^{(1)}) \\ - \sum_{i=1}^{N_1} D_{i1}^{(1)} \mathbf{\Lambda}_i^{(1)} + \mathbf{m}_1^{(1)} \nabla_{\mathbf{X}_1^{(1)}} \mathbf{h}^T(\mathbf{X}_1^{(1)}, \mathbf{U}_1^{(1)}) = 0 \end{aligned} \quad (34)$$

$$\begin{aligned} \frac{h_k}{2} \omega_j^{(k)} \nabla_{\mathbf{X}_j^{(k)}} F(\mathbf{X}_j^{(k)}, \mathbf{U}_j^{(k)}) + \frac{h_k}{2} \mathbf{\Lambda}_j^{(k)} \nabla_{\mathbf{X}_j^{(k)}} \mathbf{f}^T(\mathbf{X}_j^{(k)}, \mathbf{U}_j^{(k)}) \\ - \sum_{i=1}^{N_k} D_{ij}^{(k)} \mathbf{\Lambda}_i^{(k)} + \mathbf{m}_j^{(k)} \nabla_{\mathbf{X}_j^{(k)}} \mathbf{h}^T(\mathbf{X}_j^{(k)}, \mathbf{U}_j^{(k)}) = 0 \\ (1 \leq k \leq K, 1 \leq j \leq N_k, \mathbf{X}_j^{(k)} \neq \mathbf{X}_1^{(1)}) \end{aligned} \quad (35)$$

$$\begin{aligned} \frac{h_k}{2} \omega_j^{(k)} \nabla_{\mathbf{U}_j^{(k)}} F(\mathbf{X}_j^{(k)}, \mathbf{U}_j^{(k)}) + \frac{h_k}{2} \mathbf{\Lambda}_j^{(k)} \nabla_{\mathbf{U}_j^{(k)}} \mathbf{f}^T(\mathbf{X}_j^{(k)}, \mathbf{U}_j^{(k)}) \\ + \mathbf{m}_j^{(k)} \nabla_{\mathbf{U}_j^{(k)}} \mathbf{h}^T(\mathbf{X}_j^{(k)}, \mathbf{U}_j^{(k)}) = 0 \\ (1 \leq k \leq K, 1 \leq j \leq N_k) \end{aligned}$$

Let

$$\boldsymbol{\lambda}_i^{(k)} = \frac{\mathbf{\Lambda}_i^{(k)}}{\omega_i^{(k)}}, \boldsymbol{\mu}_i^{(k)} = \frac{\mathbf{m}_i^{(k)}}{\omega_i^{(k)}}, \mathbf{v}^{(k)} = \boldsymbol{\xi}^{(k)} \quad (37)$$

$$\boldsymbol{\lambda}_{N_K+1}^{(K)} = \left( \mathbf{D}_{N_K+1}^{(K)} \right)^T \cdot \mathbf{\Lambda}^{(K)} \quad (38)$$

$$D_{11}^{+(1)} = -D_{11}^{(1)} - \frac{1}{\omega_1^{(1)}} \quad (39)$$

$$D_{ij}^{+(k)} = -\frac{\omega_j^{(k)}}{\omega_i^{(k)}} D_{ji}^{(k)}, D_{ji}^{(k)} \neq D_{11}^{(1)} \quad (40)$$

Combining Eqs. (37) ~ (40), rewrite the first-order necessary conditions (33) ~ (36) into the following form

$$\begin{aligned} \nabla_{\mathbf{X}_{N_K+1}^{(K)}} \left( E(\mathbf{X}_1^{(1)}, \mathbf{X}_{N_K+1}^{(K)}) + \mathbf{v}^{(k)} \cdot \mathbf{e}^T (\mathbf{X}_1^{(1)}, \mathbf{X}_{N_K+1}^{(K)}) \right) \\ - \boldsymbol{\lambda}_{N_K+1}^{(K)} = 0 \end{aligned} \quad (41)$$

$$\begin{aligned} \sum_{j=1}^{N_1} D_{1j}^{+(1)} \boldsymbol{\lambda}_j^{(1)} + \frac{h_1}{2} \nabla_{\mathbf{X}_1^{(1)}} (F + \boldsymbol{\lambda}_1^{(1)} \mathbf{f}^T) \\ + \frac{1}{\omega_1^{(1)}} \left( \nabla_{\mathbf{X}_1^{(1)}} (E + \mathbf{v}^{(k)} \cdot \mathbf{e}^T) + \boldsymbol{\lambda}_1^{(1)} \right) \\ + \boldsymbol{\mu}_1^{(1)} \nabla_{\mathbf{X}_1^{(1)}} \mathbf{h}^T(\mathbf{X}_1^{(1)}, \mathbf{U}_1^{(1)}) = 0 \end{aligned} \quad (42)$$

$$\begin{aligned} \sum_{j=1}^{N_k} D_{ij}^{+(k)} \boldsymbol{\lambda}_j^{(k)} + \frac{h_k}{2} \nabla_{\mathbf{X}_i^{(k)}} (F(\mathbf{X}_i^{(k)}, \mathbf{U}_i^{(k)}) \\ + \boldsymbol{\lambda}_i^{(k)} \mathbf{f}^T(\mathbf{X}_i^{(k)}, \mathbf{U}_i^{(k)})) + \boldsymbol{\mu}_i^{(k)} \nabla_{\mathbf{X}_i^{(k)}} \mathbf{h}^T(\mathbf{X}_i^{(k)}, \mathbf{U}_i^{(k)}) = 0 \\ (1 \leq k \leq K, 1 \leq i \leq N_k, \mathbf{X}_i^{(k)} \neq \mathbf{X}_1^{(1)}) \end{aligned} \quad (43)$$

$$\begin{aligned} \frac{h_k}{2} \nabla_{\mathbf{U}_i^{(k)}} (F(\mathbf{X}_i^{(k)}, \mathbf{U}_i^{(k)}) + \boldsymbol{\lambda}_i^{(k)} \mathbf{f}^T(\mathbf{X}_i^{(k)}, \mathbf{U}_i^{(k)})) \\ + \boldsymbol{\mu}_i^{(k)} \nabla_{\mathbf{U}_i^{(k)}} \mathbf{h}^T(\mathbf{X}_i^{(k)}, \mathbf{U}_i^{(k)}) = 0 \quad (1 \leq k \leq K, 1 \leq i \leq N_k) \end{aligned} \quad (44)$$

The proof of Theorem 1 requires the following Proposition [45].

**Proposition 1** [45] If Assumption 1 ~ Assumption 3 are established. For  $\boldsymbol{\theta}^* \in \Xi$  and  $r > 0$ , let the mapping  $\Gamma: \Xi \mapsto \Psi$

be continuously Frechet differentiable within  $B_r(\boldsymbol{\theta}^*)$ . Where  $\Xi$  is a Banach space and  $\Psi$  is a normed linear space. If the following conditions are true

(C1)  $\nabla\Gamma(\boldsymbol{\theta}^*)$  is invertible.

(C2) For any  $\sigma > 0$ , there is  $r > 0$  such that  $\|\nabla\Gamma(\boldsymbol{\theta}) - \nabla\Gamma(\boldsymbol{\theta}^*)\| \leq \sigma$  holds for all  $\boldsymbol{\theta} \in B_r(\boldsymbol{\theta}^*)$ . Let  $\varpi = \|\nabla\Gamma(\boldsymbol{\theta}^*)^{-1}\|$ , if there is  $\sigma\varpi < 1$  and  $\|\Gamma(\boldsymbol{\theta}^*)\| \leq (1 - \sigma\varpi)r/\varpi$ . Then there is unique  $\boldsymbol{\theta} \in B_r(\boldsymbol{\theta}^*)$  such that  $\Gamma(\boldsymbol{\theta}) = 0$ , and at the same time there is

$$\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_\infty \leq \frac{\varpi}{1 - \sigma\varpi} \|\Gamma(\boldsymbol{\theta}^*)\|_\infty \quad (45)$$

Let  $\boldsymbol{\theta} = (X^{N_k}, U^{N_k}, \boldsymbol{\lambda}^{N_k}, \boldsymbol{\mu}^{N_k}, \mathbf{v}^{(k)})$  and  $\boldsymbol{\theta}^* = (X^{*(k)}, U^{*(k)}, \boldsymbol{\lambda}^{*(k)}, \boldsymbol{\mu}^{*(k)}, \mathbf{v}^{*(k)})$ . At the same time, combining Eqs. (15) and (41)~(44), the expression of mapping  $\Gamma$  at  $\boldsymbol{\theta}^*$  is defined as  $\Gamma(\boldsymbol{\theta}^*) = (\Gamma_1(\boldsymbol{\theta}^*), \dots, \Gamma_9(\boldsymbol{\theta}^*))^T$ , where

$$\Gamma_1(\boldsymbol{\theta}^*) = \nabla_{X_{N_k+1}^{*(K)}} (E + \mathbf{v}^{*(k)} \cdot \mathbf{e}^T) - \boldsymbol{\lambda}_{N_k+1}^{*(K)} \quad (46)$$

$$\begin{aligned} \Gamma_2(\boldsymbol{\theta}^*) = & \sum_{j=1}^{N_1} D_{1j}^{+(1)} \boldsymbol{\lambda}_j^{*(1)} + \frac{h_1}{2} \nabla_{X_1^{*(1)}} (F + \boldsymbol{\lambda}_1^{*(1)} f^T) \\ & + \boldsymbol{\mu}_1^{*(1)} \nabla_{X_1^{*(1)}} \mathbf{h}^T + \frac{1}{\omega_1^{(1)}} \left( \nabla_{X_1^{*(1)}} (E + \mathbf{v}^{*(k)} \cdot \mathbf{e}^T) + \boldsymbol{\lambda}_1^{*(1)} \right) \end{aligned} \quad (47)$$

$$\begin{aligned} \Gamma_3(\boldsymbol{\theta}^*) = & \sum_{j=1}^{N_k} D_{ij}^{+(k)} \boldsymbol{\lambda}_j^{*(k)} + \frac{h_k}{2} \nabla_{X_i^{*(k)}} (F + \boldsymbol{\lambda}_i^{*(k)} f^T) \\ & + \boldsymbol{\mu}_i^{*(k)} \nabla_{X_i^{*(k)}} \mathbf{h}^T \quad (1 \leq k \leq K, 1 \leq i \leq N_k, X_i^{*(k)} \neq X_1^{*(1)}) \end{aligned} \quad (48)$$

$$\Gamma_4(\boldsymbol{\theta}^*) = \frac{h_k}{2} \nabla_{U_i^{*(k)}} (F + \boldsymbol{\lambda}_i^{*(k)} f^T) + \boldsymbol{\mu}_i^{*(k)} \nabla_{U_i^{*(k)}} \mathbf{h}^T \quad (1 \leq k \leq K, 1 \leq i \leq N_k) \quad (49)$$

$$\Gamma_5(\boldsymbol{\theta}^*) = \boldsymbol{\mu}_i^{*(k)} \mathbf{h}^T (X_i^{*(k)}, U_i^{*(k)}) \quad (1 \leq k \leq K, 1 \leq i \leq N_k) \quad (50)$$

$$\begin{aligned} \Gamma_6(\boldsymbol{\theta}^*) = & \sum_{j=1}^{N_k+1} X_j^{*(k)} D_{ij}^{(k)} - \frac{h_k}{2} f(X_i^{*(k)}, U_i^{*(k)}) \\ & (i = 1, 2, \dots, N_k \quad k = 1, 2, \dots, K) \end{aligned} \quad (51)$$

$$\Gamma_7(\boldsymbol{\theta}^*) = \mathbf{e} (X_1^{*(1)}, X_{N_k+1}^{*(K)}) \quad (52)$$

$$\Gamma_8(\boldsymbol{\theta}^*) = \mathbf{h} (X_i^{*(k)}, U_i^{*(k)}) \quad (53)$$

$$\Gamma_9(\boldsymbol{\theta}^*) = X_{N_k+1}^{*(k)} - X_1^{*(k+1)} \quad (54)$$

From [46], we know that the conditions C1 and C2 in Proposition 1 are established. Next, consider the size of  $\|\Gamma(\boldsymbol{\theta}^*)\|_\infty$ .

According to Assumption 1,  $\boldsymbol{\theta}^*$  satisfies Eq. (1) and Eq. (2), so for  $\Gamma(\boldsymbol{\theta}^*)$  defined above, there is

$$\begin{aligned} \Gamma_1(\boldsymbol{\theta}^*) = \Gamma_4(\boldsymbol{\theta}^*) = \Gamma_5(\boldsymbol{\theta}^*) = \Gamma_7(\boldsymbol{\theta}^*) = \Gamma_9(\boldsymbol{\theta}^*) = \mathbf{0} \\ \Gamma_8(\boldsymbol{\theta}^*) \leq \mathbf{0} \end{aligned} \quad (55)$$

Next, in order to continue to examine the size of  $\|\Gamma_2(\boldsymbol{\theta}^*)\|_\infty$ ,  $\|\Gamma_3(\boldsymbol{\theta}^*)\|_\infty$  and  $\|\Gamma_6(\boldsymbol{\theta}^*)\|_\infty$ , first introduce the following lemmas [46]

**Lemma 1** If the Lebesgue constant  $\ell_N$  is defined as

$$\ell_N = \max_{\tau \in [-1, 1]} \sum_{j=1}^{N+1} |l_j(\tau)| \quad (56)$$

Here, the definition of Lagrange interpolation polynomial  $l_j(\tau)$  is shown in Eq. (16). Then there is  $\ell_N = O(\log N)$ .

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

$$\ell_{N-1} = \max_{\tau \in [-1, 1]} \sum_{j=1}^N |l_j(\tau)| \quad (57)$$

The definition of Lagrange interpolation polynomial  $l_j(\tau)$  is shown in Eq. (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 Eq. (16), then

$$\|y - y^N\|_\infty \leq (1 + 2N^2 \ell_N) \inf_{q \in P_{N-1}} \|y - q\|_\infty \quad (58)$$

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

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

$$\inf_{q \in P_N} \|y - q\|_\infty \leq \frac{c_l}{N^l} \left\| \frac{d^l y}{dt^l} \right\|_\infty \quad (59)$$

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

**Lemma 5** According to the definition in Eq. (39) and (40), matrix  $\mathbf{D}^{+(k)} = [D_{ij}^{+(k)}]_{N_k \times N_k}$  is a differentiation matrix. That is, if  $p$  is a polynomial of degree  $N_k - 1$  and the  $i$ th element of vector  $\mathbf{P} \in \mathbf{R}^{N_k}$  is  $\mathbf{P}_i = p_i = p(\tau_i)$ , then there is

$$(\mathbf{D}^{+(k)} \mathbf{P})_i = \dot{p}(\tau_i), \quad 1 \leq i \leq N_k \quad (60)$$

The proof of Lemma 1 ~ Lemma 5 can be found in [46].

In order to analyze the size of  $\|\Gamma_3(\boldsymbol{\theta}^*)\|_\infty$ , combining with Eq. (2), we can know that in Eq. (48),  $\boldsymbol{\mu}_i^{*(k)} \nabla_{\mathbf{X}_i^{*(k)}} \mathbf{h}^T = 0$ ,  $\frac{h_k}{2} \nabla_{\mathbf{X}_i^{*(k)}} (F + \boldsymbol{\lambda}_i^{*(k)} \mathbf{f}^T) = -\dot{\boldsymbol{\lambda}}^{*(k)}(\tau_i)$ . Therefore, Eq. (47) can be simplified to

$$\Gamma_3(\boldsymbol{\theta}^*) = \sum_{j=1}^{N_k} D_{ij}^{+(k)} \boldsymbol{\lambda}_j^{*(k)} - \dot{\boldsymbol{\lambda}}^{*(k)}(\tau_i) \quad (61)$$

From Lemma 5,

$$\sum_{j=1}^{N_k} D_{ij}^{+(k)} \boldsymbol{\lambda}_j^{*(k)} = [\dot{\boldsymbol{\lambda}}^{*(k)}(\tau_i)]^{N_k-1} \quad (62)$$

Where, the superscript  $N_k - 1$  denotes that  $\boldsymbol{\lambda}^{*(k)}$  is a polynomial of order  $N_k - 1$  obtained by lagrange interpolation. Combining Eq. (61) and Eq. (62), there is

$$\Gamma_3(\boldsymbol{\theta}^*) = [\dot{\boldsymbol{\lambda}}^{*(k)}(\tau_i)]^{N_k-1} - \dot{\boldsymbol{\lambda}}^{*(k)}(\tau_i) \quad (63)$$

Let  $[\dot{\boldsymbol{\lambda}}^{*(k)}(\tau_i)]_j^{N_k-1}$  denote the  $j$ th element of vector  $[\dot{\boldsymbol{\lambda}}^{*(k)}(\tau_i)]^{N_k-1}$ , and  $\dot{\boldsymbol{\lambda}}_j^{*(k)}(\tau_i)$  denote the  $j$ th element of  $\dot{\boldsymbol{\lambda}}^{*(k)}(\tau_i)$ . there is,

$$\left| [\dot{\boldsymbol{\lambda}}^{*(k)}(\tau_i)]_j^{N_k-1} - \dot{\boldsymbol{\lambda}}_j^{*(k)}(\tau_i) \right| \leq \left\| [\dot{\boldsymbol{\lambda}}^{*(k)}]_j^{N_k-1} - \dot{\boldsymbol{\lambda}}_j^{*(k)} \right\|_\infty \quad (64)$$

According to Lemma 2, Lemma 3 and Lemma 4, we can get

$$\begin{aligned} & \left\| [\dot{\boldsymbol{\lambda}}^{*(k)}]_j^{N_k-1} - \dot{\boldsymbol{\lambda}}_j^{*(k)} \right\|_\infty \\ & \leq \left( 1 + 2(N_k - 1)^2 O\left(N_k^{\frac{1}{2}}\right) \right) \inf_{q \in P_{N_k-2}} \left\| \dot{\boldsymbol{\lambda}}_j^{*(k)} - q \right\|_\infty \\ & \leq \left( 1 + 2(N_k - 1)^2 O\left(N_k^{\frac{1}{2}}\right) \right) \frac{c_{l_k}}{N_k^{l_k}} \left\| \frac{d^{l_k+1} \boldsymbol{\lambda}_j^{*(k)}}{d\tau^{l_k+1}} \right\|_\infty \\ & \leq \frac{c_{l_k}}{N_k^{l_k - \frac{5}{2}}} \cdot \left( \frac{h_k}{2} \right)^{l_k+1} \left\| \frac{d^{l_k+1} \boldsymbol{\lambda}_j^{*(k)}}{d\tau^{l_k+1}} \right\|_\infty \\ & \leq \frac{c_{l_k} h_k^{l_k+1}}{N_k^{l_k - \frac{5}{2}}} \end{aligned} \quad (65)$$

Since Eqs. (64) and (65) hold for any  $i$  and  $j$ , there is

$$\|\Gamma_3(\boldsymbol{\theta}^*)\|_\infty \leq \frac{c_{l_k} h_k^{l_k+1}}{N_k^{l_k - \frac{5}{2}}} \quad (66)$$

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

$$\|\Gamma_2(\boldsymbol{\theta}^*)\|_\infty \leq \frac{c_{l_k} h_k^{l_k+1}}{N_k^{l_k - \frac{5}{2}}} \quad (67)$$

Finally, in order to analyze the size of  $\|\Gamma_6(\boldsymbol{\theta}^*)\|_\infty$ , we know  $\frac{h_k}{2} f(\mathbf{X}_i^{*(k)}, \mathbf{U}_i^{*(k)}) = \dot{\mathbf{x}}^{*(k)}(\tau_i)$  according to Eq. (1), and because  $\mathbf{D}^{(k)} = [D_{ij}^{(k)}]$  is a differentiation matrix, Eq. (51) can be written as

$$\Gamma_6(\boldsymbol{\theta}^*) = [\dot{\mathbf{x}}^{*(k)}(\tau_i)]^{N_k} - \dot{\mathbf{x}}^{*(k)}(\tau_i) \quad (68)$$

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

$$\left| [\dot{\mathbf{x}}^{*(k)}(\tau_i)]_j^{N_k} - \dot{\mathbf{x}}_j^{*(k)}(\tau_i) \right| \leq \left\| [\dot{\mathbf{x}}^{*(k)}]_j^{N_k} - \dot{\mathbf{x}}_j^{*(k)} \right\|_\infty \quad (69)$$

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

$$\begin{aligned}
& \left\| \left[ \mathbf{x}^{*(k)} \right]_j^{N_k} - \mathbf{x}_j^{*(k)} \right\|_{\infty} \\
& \leq (1 + 2N_k^2 O(\log N)) \inf_{q \in P_{N_k-1}} \left\| \dot{\mathbf{x}}_j^{*(k)} - q \right\|_{\infty} \\
& \leq (1 + 2N_k^2 O(\log N)) \frac{c_{l_k}}{N_k^{l_k}} \left\| \frac{d^{l_k+1} \mathbf{x}_j^{*(k)}}{d\tau^{l_k+1}} \right\|_{\infty} \quad (70) \\
& \leq \frac{c_{l_k} \log N}{N_k^{l_k-2}} \cdot \left( \frac{h_k}{2} \right)^{l_k+1} \left\| \frac{d^{l_k+1} \mathbf{x}_j^{*(k)}}{d\tau^{l_k+1}} \right\|_{\infty} \\
& \leq \frac{c_{l_k} h_k^{l_k+1} \log N}{N_k^{l_k-2}} \leq \frac{c_{l_k} h_k^{l_k+1}}{N_k^{l_k-\frac{5}{2}}}
\end{aligned}$$

Because Eqs. (69) and (70) hold for any  $i$  and  $j$ , we get

$$\| \Gamma_6(\boldsymbol{\theta}^*) \|_{\infty} \leq \frac{c_{l_k} h_k^{l_k+1}}{N_k^{l_k-\frac{5}{2}}} \quad (71)$$

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

$$\| \Gamma(\boldsymbol{\theta}^*) \|_{\infty} \leq \frac{c_{l_k} h_k^{l_k+1}}{N_k^{l_k-\frac{5}{2}}} \quad (72)$$

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

$$\begin{aligned}
\| \boldsymbol{\theta} - \boldsymbol{\theta}^* \|_{\infty} &= \max \left\{ \left\| \mathbf{X}^{N_k} - \mathbf{X}^{*(k)} \right\|_{\infty}, \left\| \mathbf{U}^{N_k} - \mathbf{U}^{*(k)} \right\|_{\infty}, \right. \\
& \quad \left. \left\| \boldsymbol{\lambda}^{N_k} - \boldsymbol{\lambda}^{*(k)} \right\|_{\infty}, \left\| \boldsymbol{\mu}^{N_k} - \boldsymbol{\mu}^{*(k)} \right\|_{\infty}, \left\| \mathbf{v}^{(k)} - \mathbf{v}^{*(k)} \right\|_{\infty} \right\} \\
&\leq \frac{\boldsymbol{\varpi}}{1 - \boldsymbol{\varpi} \sigma} \| \Gamma(\boldsymbol{\theta}^*) \|_{\infty} \\
&\leq \frac{\boldsymbol{\varpi}}{1 - \boldsymbol{\varpi} \sigma} \frac{c_{l_k} h_k^{l_k+1}}{N_k^{l_k-\frac{5}{2}}} \quad (73)
\end{aligned}$$

Since  $\boldsymbol{\varpi} / (1 - \boldsymbol{\varpi} \sigma) > 0$  is bounded, let  $C_{l_k} = \boldsymbol{\varpi} c_{l_k} / (1 - \boldsymbol{\varpi} \sigma)$ . In conclusion, Theorem 1 is established.  $\square$

## 5 Example and Discussion

We verify the effectiveness of the improved adaptive hp algorithm through several groups of simulation examples [47–49], 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 Matlab2014a. In addition, the algorithm proposed in this paper is developed based on the open source hp pseudospectral optimization software GPOPS [50], and the NLP solver used is SNOPT [9].

### 5.1 Example 1: Soft Lunar Landing Problem

The optimal control problem for soft landing on the moon was proposed in [47], and it is restated here as follows:

Find the optimal control  $u$  such that

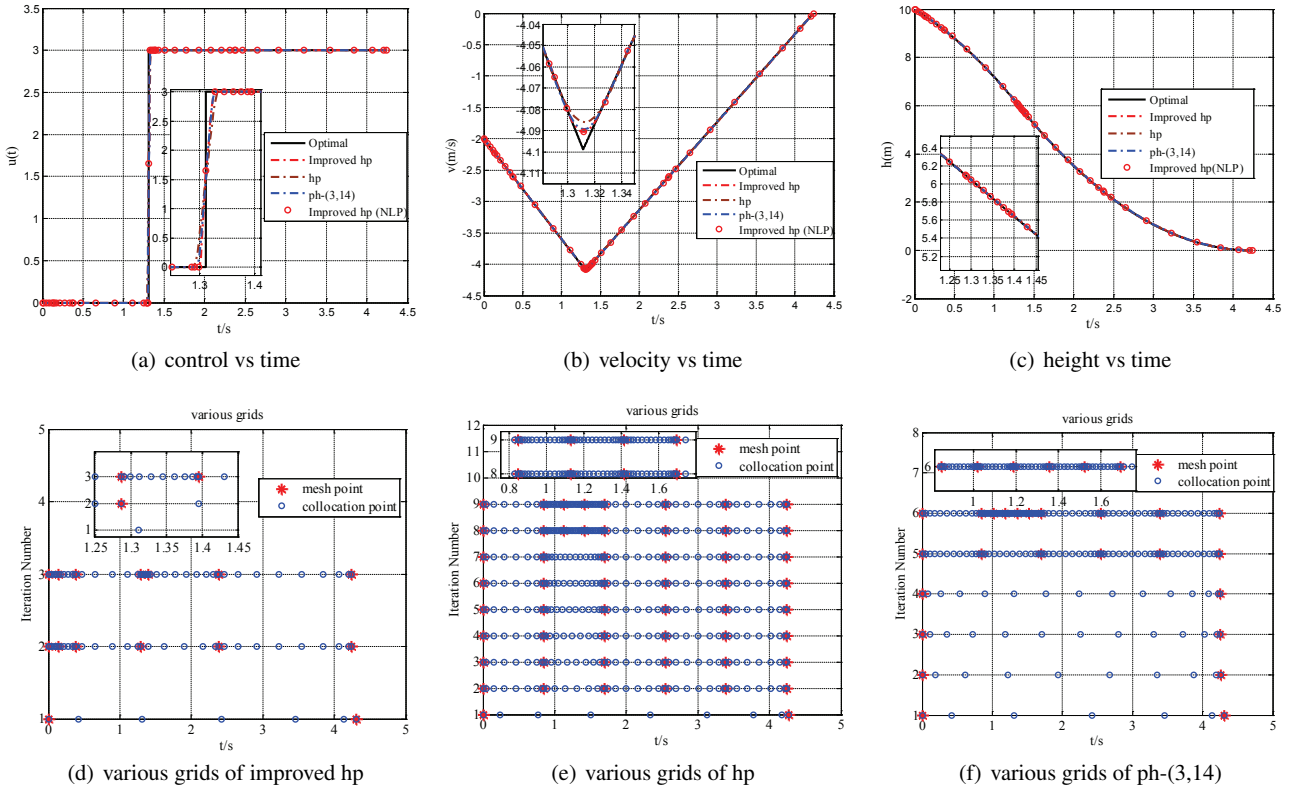
$$\begin{cases} \min J = \int_0^{t_f} u \, dt \\ \text{s.t. } \dot{h} = v \\ \dot{v} = -g + u \\ h(0) = 10, h(t_f) = 0 \\ v(0) = -2, v(t_f) = 0 \\ 0 \leq u \leq 3 \end{cases} \quad (74)$$

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

$$u^*(t) = \begin{cases} 0 & 0 \leq t \leq t_s^* \\ 3 & t_s^* < t \leq t_f^* \end{cases} \quad (75)$$

Combining with the initial conditions set by Eq. (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:  $\varepsilon = 10^{-6}$ ,  $\eta = 1$ ,  $\rho = 3$ . 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  $\varepsilon$  of mesh accuracy, the hp method in [25] and the ph- $(N_{\min}, N_{\max})$  method in [22] are used to solve the problem and compare with the method in this paper. The notation



**Fig. 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

ph- $(N_{\min}, N_{\max})$  here represents the range of the number of collocation points of the ph method from  $N_{\min}$  to  $N_{\max}$ .

As shown in Fig. 5(a)~(c), all three methods can accurately solve the problem under the same tolerance range  $\varepsilon$  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  $t=1.3117s$ , 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 Fig. 5(e) and (f)). As shown in Fig. 5(d), 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 0.14s, significantly less than the other two methods).

## 5.2 Example 2: Hyper-sensitive Problem

Consider the Hyper-sensitive optimal control problem mentioned in [48] and restate it as follows:

$$\begin{cases} \min J = \frac{1}{2} \int_0^{t_f} (x^2 + u^2) dt \\ s.t. \dot{x} = -x^3 + u \\ x(0) = 1 \\ x(t_f) = 1 \end{cases} \quad (76)$$

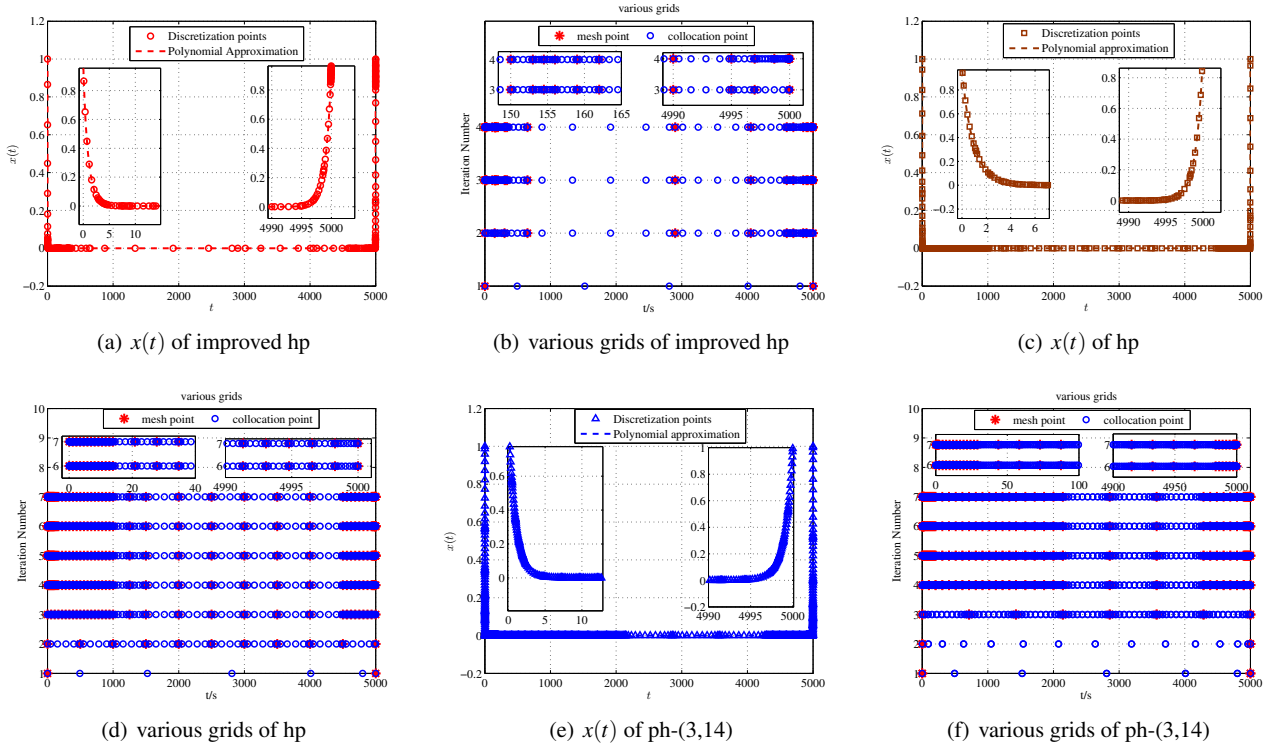


Fig. 6 Optimization results of example 2

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

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:  $\varepsilon = 10^{-6}$ ,  $\eta = 1$ ,  $\rho = 3$ . Similarly, the hp method [25] and the ph- $(N_{\min}, N_{\max})$  method [22] are compared with the method in this paper.

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  $x(t)$  curve (see Fig. 6(a), (c) and (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 Fig. 6(f)), which affects the efficiency of the solving. The reason is that when the method is performing mesh re-

finement, 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 Fig. 6(d)), 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 Fig. 6(b)). Table 2 also shows the same conclusion. As shown in 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 0.14s).

### 5.3 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 [49]

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

$$\left\{ \begin{array}{l} \min J = \beta(t_f) \\ s.t. \dot{x} = V \cos \gamma \sin \Psi + W_x \\ \dot{y} = V \cos \gamma \cos \Psi \\ \dot{h} = V \sin \gamma \\ m\dot{V} = -D - mg \sin \gamma - m\dot{W}_x \cos \gamma \sin \Psi \\ mV\dot{\gamma} = L \cos \sigma - mg \cos \gamma + m\dot{W}_x \sin \gamma \sin \Psi \\ mV \cos \gamma \dot{\Psi} = L \sin \sigma - m\dot{W}_x \cos \Psi \\ \dot{W}_x = \beta V \sin \gamma, \beta = W_x/h \\ (x(0), y(0), h(0)) = (0, 0, 0) \\ (x(t_f), y(t_f), h(t_f), V(t_f), \gamma(t_f), \Psi(t_f)) \\ = (x(0), y(0), h(0), V(0), \gamma(0), \Psi(0) - 2\pi) \\ -2 \leq L/(mg) \leq 5 \end{array} \right. \quad (77)$$

Here, lift  $L$  and drag  $D$  are expressed as

$$\left\{ \begin{array}{l} L = qSC_L \\ D = qSC_D \\ q = \rho V^2/2 \\ C_D = C_{D0} + KC_L^2 \end{array} \right. \quad (78)$$

The definitions and values of related variables and parameters involved in Eqs. (77) and (78) are detailed in [23].

In order to find out the optimal control  $(C_L, \sigma)$  satisfying Eq. (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:  $\varepsilon = 10^{-6}$ ,  $\eta = 1$ ,  $\rho = 3$ . At the same time, the hp method [25] and the ph- $(N_{\min}, N_{\max})$  method [22] are used to solve the problem, which is compared with the method in this paper.

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 Fig. 7(a)~(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 Fig. 7(e)~(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 4.24s).

## 6 Conclusion

In order to improve the efficiency of solving 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 grid iterations and shortens the calculation time. Simulation results also show that the number of grid iterations, the number of required collocation points and the CPU time of the proposed algorithm are less than other methods.

## 7 Acknowledgements

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

**Conflict of Interest:** The authors declare that they have no conflict of interest.

**Funding:** This work was supported by the National Natural Science Foundation of China (Grant no. 61773398 and no. 61873278).

**Data availability statements :** All data generated or analysed during this study are included in this published article (and its supplementary information files).

## References

1. D. A. Benson, G. T. Huntington, and T. P. Thorvaldsen, "Direct trajectory optimization and costate estimation via an orthogonal collocation method," *Journal of Guidance Control and Dynamics*, vol. 29, no. 6, pp. 1435-1440, Nov-Dec, 2006.
2. G. T. Huntington, and A. V. Rao, "Optimal reconfiguration of spacecraft formations using the Gauss pseudospectral method," *Journal of Guidance Control and Dynamics*, vol. 31, no. 3, pp. 689-698, May-Jun, 2008.
3. Q. Gong, I. M. Ross, and W. Kang, "Connections between the covector mapping theorem and convergence of pseudospectral methods for optimal control," *Computational Optimization and Applications*, vol. 41, no. 3, pp. 307-335, Dec, 2008.

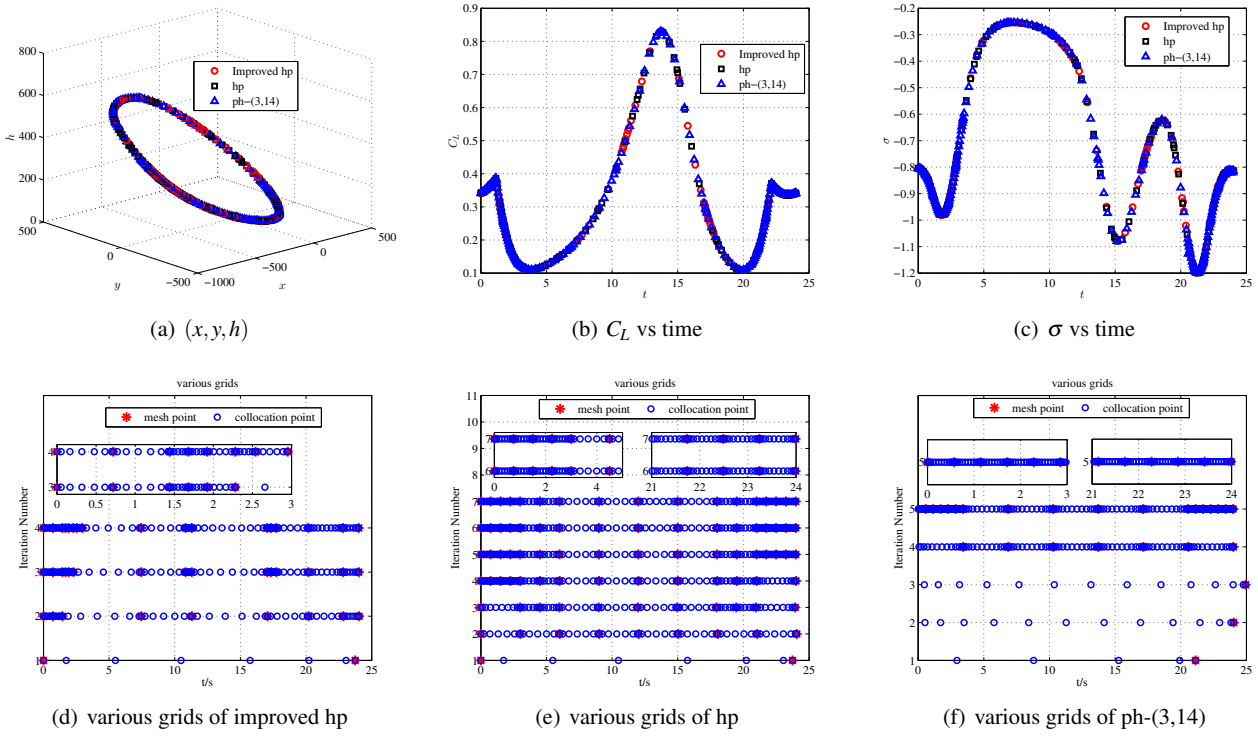


Fig. 7 Optimization results of example 3

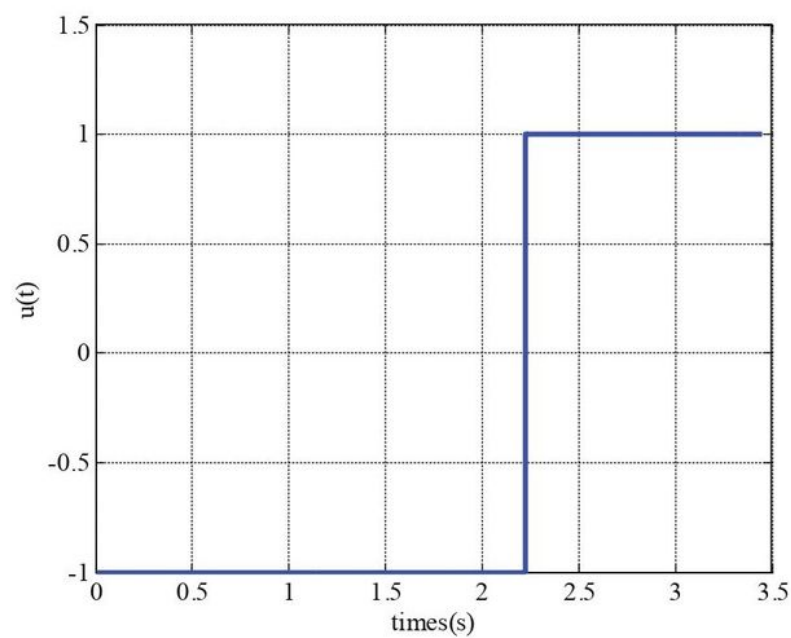
Table 3 Comparison of mesh refinement results of example 3

Method	CPU time (s)	Total number of collocation points	Number of mesh intervals	Number of grid iterations
Improved hp	4.24	123	16	3
hp	7.11	148	16	6
ph-(3,14)	8.56	239	17	4
ph-(5,16)	8.81	257	16	5
ph-(6,18)	18.33	451	25	5

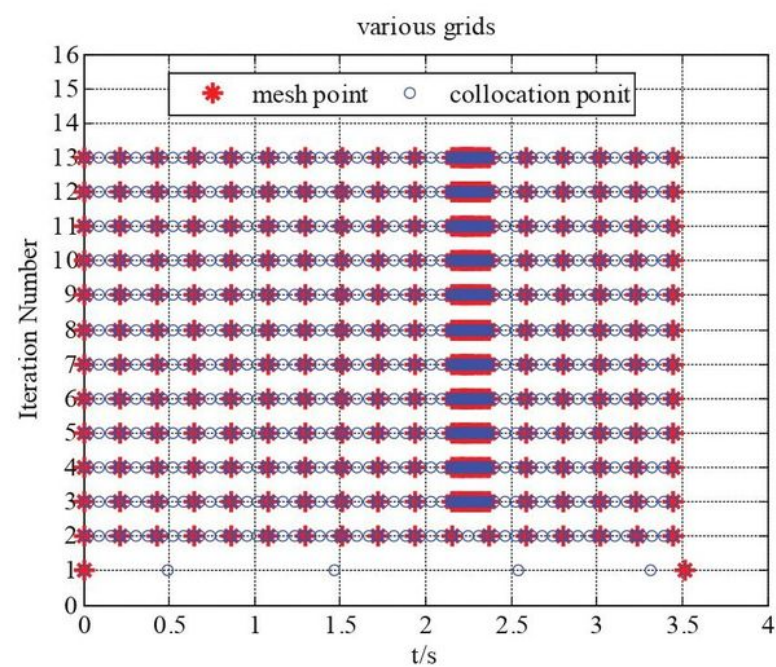
- D. Garg, M. Patterson, and W. W. Hager, "A unified framework for the numerical solution of optimal control problems using pseudospectral methods," *Automatica*, vol. 46, no. 11, pp. 1843-1851, Nov, 2010.
- D. Garg, M. A. Patterson, and C. Francolin, "Direct trajectory optimization and costate estimation of finite-horizon and infinite-horizon optimal control problems using a Radau pseudospectral method," *Computational Optimization and Applications*, vol. 49, no. 2, pp. 335-358, Jun, 2011.
- D. Garg, W. W. Hager, and A. V. Rao, "Pseudospectral methods for solving infinite-horizon optimal control problems," *Automatica*, vol. 47, no. 4, pp. 829-837, Apr, 2011.
- M. A. Patterson, and A. V. Rao, "Exploiting Sparsity in Direct Collocation Pseudospectral Methods for Solving Optimal Control Problems," *Journal of Spacecraft and Rockets*, vol. 49, no. 2, pp. 364-377, Mar-Apr, 2012.
- C. C. Francolin, D. A. Benson, and W. W. Hager, "Costate approximation in optimal control using in-tegral Gaussian quadrature orthogonal collocation methods," *Optimal Control Applications & Methods*, vol. 36, no. 4, pp. 381-397, Jul-Aug, 2015.
- P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization (Reprinted from SIAM Journal Optimization, vol 12, pg 979-1006, 2002)," *Siam Review*, vol. 47, no. 1, pp. 99-131, Mar, 2005.
- M. A. Patterson, and A. V. Rao, "Exploiting Sparsity in Direct Collocation Pseudospectral Methods for Solving Optimal Control Problems," *Journal of Spacecraft and Rockets*, vol. 49, no. 2, pp. 364-377, Mar-Apr, 2012.
- F. Fahroo and I. M. Ross, *Advances in Pseudospectral Methods for Optimal Control*. faculty publications, 2008.
- Q. Gong, I. M. Ross, and W. Kang, "On the pseudospectral covector mapping theorem for nonlinear optimal control," *Proceedings of the 45th IEEE Conference on Decision and Control*, Vols 1-14, pp. 2683-+, 2006.
- Q. Gong, I. M. Ross, and W. Kang, "Connections between the covector mapping theorem and convergence of pseudospectral methods for optimal control," *Computational Optimization and Applications*, vol. 41, no. 3, pp. 307-335, Dec, 2008.
- Q. Gong, W. Kang, and I. M. Ross, "A pseudospectral method for the optimal control of constrained feedback linearizable systems," *IEEE Transactions on Automatic Control*, vol. 51, no. 7, pp. 1115-1129, Jul, 2006.
- D. Benson, *A Gauss pseudospectral transcription for optimal control*, Ph.D. Thesis, Department of Aero-nautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, 2004.
- J. T. Betts, *Practical Methods for Optimal Control Using Nonlinear Programming*, SIAM Press: Philadelphia, 2001.

17. S. Jain, and P. Tsiotras, "Trajectory optimization using multiresolution techniques," *Journal of Guidance Control and Dynamics*, vol. 31, no. 5, pp. 1424-1436, Sep-Oct, 2008.
18. Y. Zhao and P. Tsiotras, *A density-function based mesh refinement algorithm for solving optimal control problems*, Infotech and Aerospace Conference, AIAA Paper 2009-2019, Seattle, Washington, April 2009.
19. M. Y. Hussaini and T. A. Zang, "Spectral Methods in Fluid-Dynamics," *Annual Review of Fluid Mechanics*, vol. 19, pp. 339-367, 1987.
20. B. Fornberg, *A Practical Guide to Pseudospectral Methods*, Cambridge University Press: Cambridge, 1998.
21. Trefethen, *Spectral Methods Using MATLAB*, SIAM Press: Philadelphia, 2000.
22. M. A. Patterson, W. W. Hager, and A. V. Rao, "A hp mesh refinement method for optimal control," *Optimal Control Applications & Methods*, vol. 36, no. 4, pp. 398-421, Jul-Aug, 2015.
23. C. L. Darby, W. W. Hager, and A. V. Rao, "An hp-adaptive pseudospectral method for solving optimal control problems," *Optimal Control Applications & Methods*, vol. 32, no. 4, pp. 476-502, Jul-Aug, 2011.
24. F. J. Liu, W. W. Hager, and A. V. Rao, "An hp Mesh Refinement Method for Optimal Control Using Discontinuity Detection and Mesh Size Reduction," *2014 IEEE 53rd Annual Conference on Decision and Control (Cdc)*, pp. 5868-5873, 2014.
25. F. J. Liu, W. W. Hager, and A. V. Rao, "Adaptive mesh refinement method for optimal control using nonsmoothness detection and mesh size reduction," *Journal of the Franklin Institute-Engineering and Applied Mathematics*, vol. 352, no. 10, pp. 4081-4106, Oct, 2015.
26. P. Daniel, A. Ern, and I. Smears, "An adaptive hp-refinement strategy with computable guaranteed bound on the error reduction factor," *Computers & Mathematics with Applications*, vol. 76, no. 5, pp. 967-983, Sep 1, 2018.
27. N. Koeppen, I. M. Ross, and L. C. Wilcox, "Fast Mesh Refinement in Pseudospectral Optimal Control," *Journal of Guidance Control and Dynamics*, vol. 42, no. 4, pp. 711-722, Apr, 2019.
28. J. S. Zhao and S. Li, "Mars atmospheric entry trajectory optimization with maximum parachute de-ployment altitude using adaptive mesh refinement," *Acta Astronautica*, vol. 160, pp. 401-413, Jul, 2019.
29. Z. W. Feng, Q. B. Zhang, and J. Q. Ge, "Mesh Adaptation Method for Optimal Control With Non-Smooth Control Using Second-Generation Wavelet," *Ieee Access*, vol. 7, pp. 135076-135086, 2019.
30. Y. M. Agamawi and A. V. Rao, "CGPOPS: A C plus plus Software for Solving Multiple-Phase Optimal Control Problems Using Adaptive Gaussian Quadrature Collocation and Sparse Nonlinear Programming," *Acm Transactions on Mathematical Software*, vol. 46, no. 3, Sep, 2020.
31. N. B. Li, H. M. Lei, and L. Shao, "Trajectory Optimization Based on Multi-Interval Mesh Refinement Method," *Mathematical Problems in Engineering*, vol. 2017.
32. L. Xiao, L. Lv, and P. Liu, "A novel adaptive Gauss pseudospectral method for nonlinear optimal control of constrained hypersonic re-entry vehicle problem," *International Journal of Adaptive Control and Signal Processing*, vol. 32, no. 9, pp. 1243-1258, Sep, 2018.
33. H. M. Lei, T. Liu, and D. Li, "Adaptive mesh refinement method for optimal control based on Hermite-Legendre-Gauss-Lobatto direct transcription," *Journal of Vibroengineering*, vol. 19, no. 8, pp. 6036-6048, Dec, 2017.
34. F. J. Liu, W. W. Hager, and A. V. Rao, "Adaptive Mesh Refinement Method for Optimal Control Using Decay Rates of Legendre Polynomial Coefficients," *Ieee Transactions on Control Systems Technology*, vol. 26, no. 4, pp. 1475-1483, Jul, 2018.
35. M. Souzban, O. S. Fard, and A. H. Borzabadi, "A rapid-based improvement on some mesh refinement strategies in solving optimal control problems," *Ima Journal of Mathematical Control and Information*, vol. 37, no. 2, pp. 351-376, Jun, 2020.
36. N. Ejlali and S. M. Hosseini, "Adaptive control parameterization method by density functions for optimal control problems," *Ima Journal of Mathematical Control and Information*, vol. 37, no. 2, pp. 497-512, Jun, 2020.
37. J. S. Zhao and S. Li, "Adaptive mesh refinement method for solving optimal control problems using interpolation error analysis and improved data compression," *Journal of the Franklin Institute-Engineering and Applied Mathematics*, vol. 357, no. 3, pp. 1603-1627, Feb, 2020.
38. H. Fujihira, "Necessary and Sufficient Conditions for Bang-Bang Control," *Journal of Optimization Theory and Applications*, vol. 25, no. 4, pp. 549-554, 1978.
39. Z. W. Lu, G. Y. Tian, and S. Onori, "Multistage Time-Optimal Control for Synchronization Process in Electric-Driven Mechanical Transmission With Angle Alignment Considering Torque Response Process," *Journal of Dynamic Systems Measurement and Control-Transactions of the Asme*, vol. 143, no. 4, Apr 1, 2021.
40. Y. R. Han, Y. Zhao, and P. Wang, "Finite-time rate anti-bump switching control for switched systems," *Applied Mathematics and Computation*, vol. 401, Jul 15, 2021.
41. R. B. Vinter, *Optimal Control*, Birkhäuser, Boston, 2000.
42. I. M. Ross and F. Fahroo, "Legendre pseudospectral approximations of optimal control problems," *New Trends in Nonlinear Dynamics and Control, and Their Applications*, vol. 295, pp. 327-342, 2003.
43. W. W. Hager, "Numerical analysis in optimal control," *Optimal Control of Complex Structures*, vol. 139, pp. 83-93, 2002.
44. M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions with Formulas Graphs, and Mathematical Tables*, Dover Publications, New York, 1965.
45. A. L. Dontchev, W. W. Hager, and V. M. Veliov, "Second-order Runge-Kutta approximations in control constrained optimal control," *Siam Journal on Numerical Analysis*, vol. 38, no. 1, pp. 202-226, Jun 23, 2000.
46. H. Hou, *Convergence analysis of orthogonal collocation methods for unconstrained optimal control*, Ph.D. Thesis, University of Florida, August 2013.
47. J. Meditch, "On the problem of optimal thrust programming for a soft lunar landing," *IEEE Transactions on Automatic Control*, vol. 9, no. 4, pp. 477-484, 1964.
48. A. V. Rao and K. D. Mease, "Eigenvector approximate dichotomic basis method for solving hyper-sensitive optimal control problems," *Optimal Control Applications & Methods*, vol. 21, no. 1, pp. 1-19, Jan-Feb, 2000.
49. Y. Y. J. Zhao, "Optimal patterns of glider dynamic soaring," *Optimal Control Applications & Methods*, vol. 25, no. 2, pp. 67-89, Mar-Apr, 2004.
50. A. V. Rao, D. A. Benson, and C. Darby, "Algorithm 902: GPOPS, A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using the Gauss Pseudospectral Method," *Acm Transactions on Mathematical Software*, vol. 37, no. 2, Apr, 2010.

# Figures



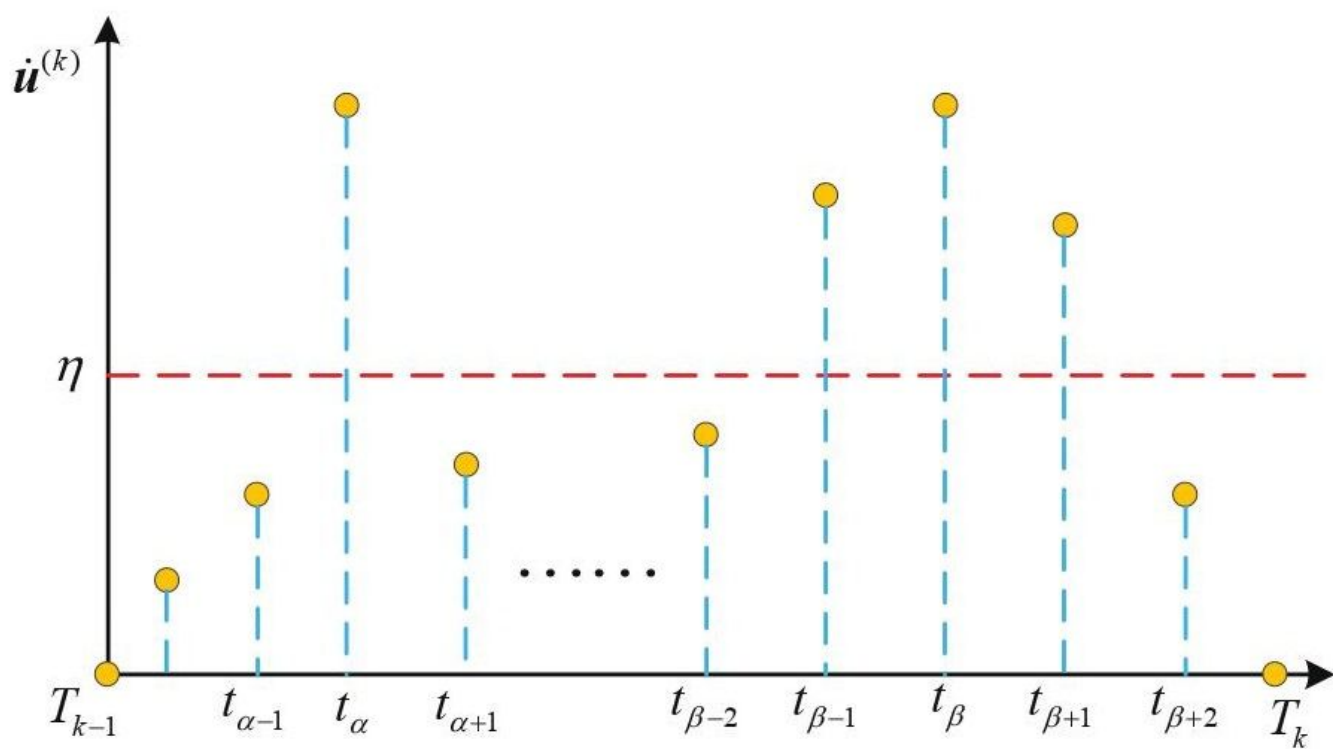
(a) control vs time



(b) approximation points of various grids

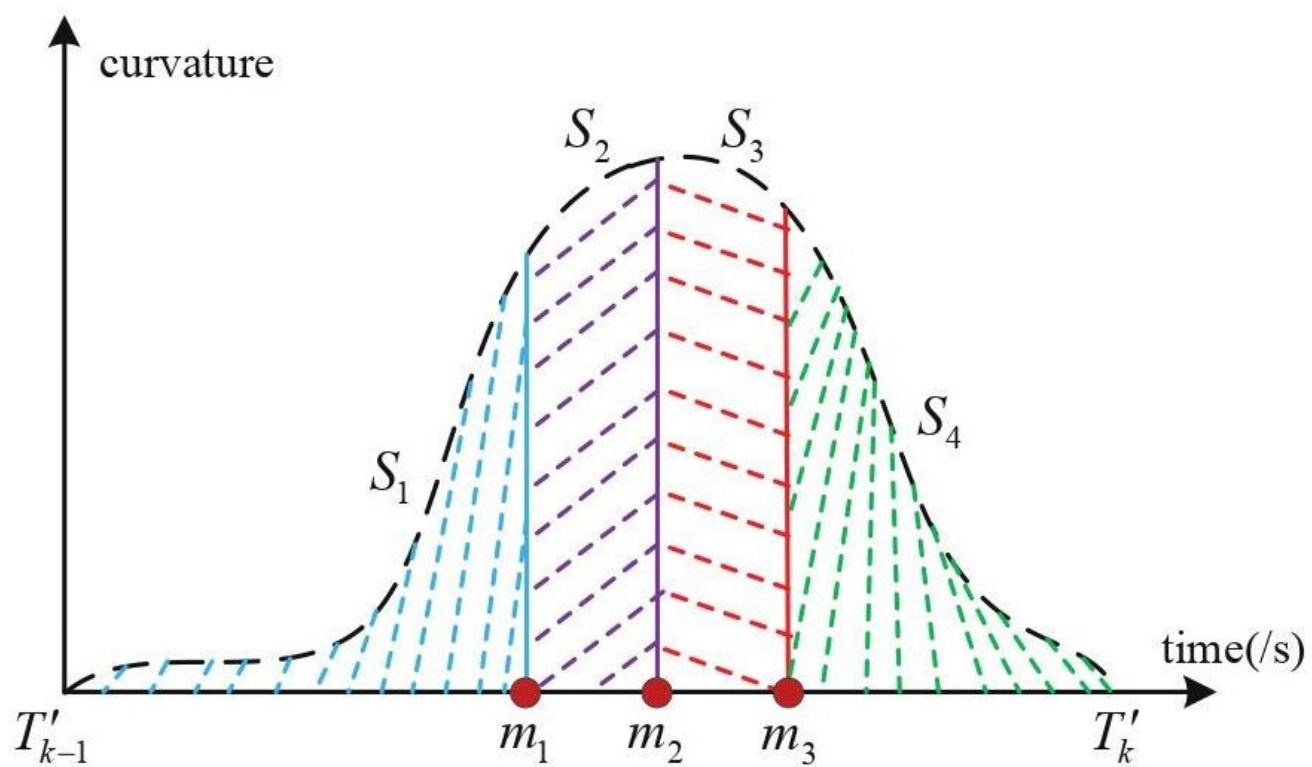
Figure 1

Optimization results of system (22)



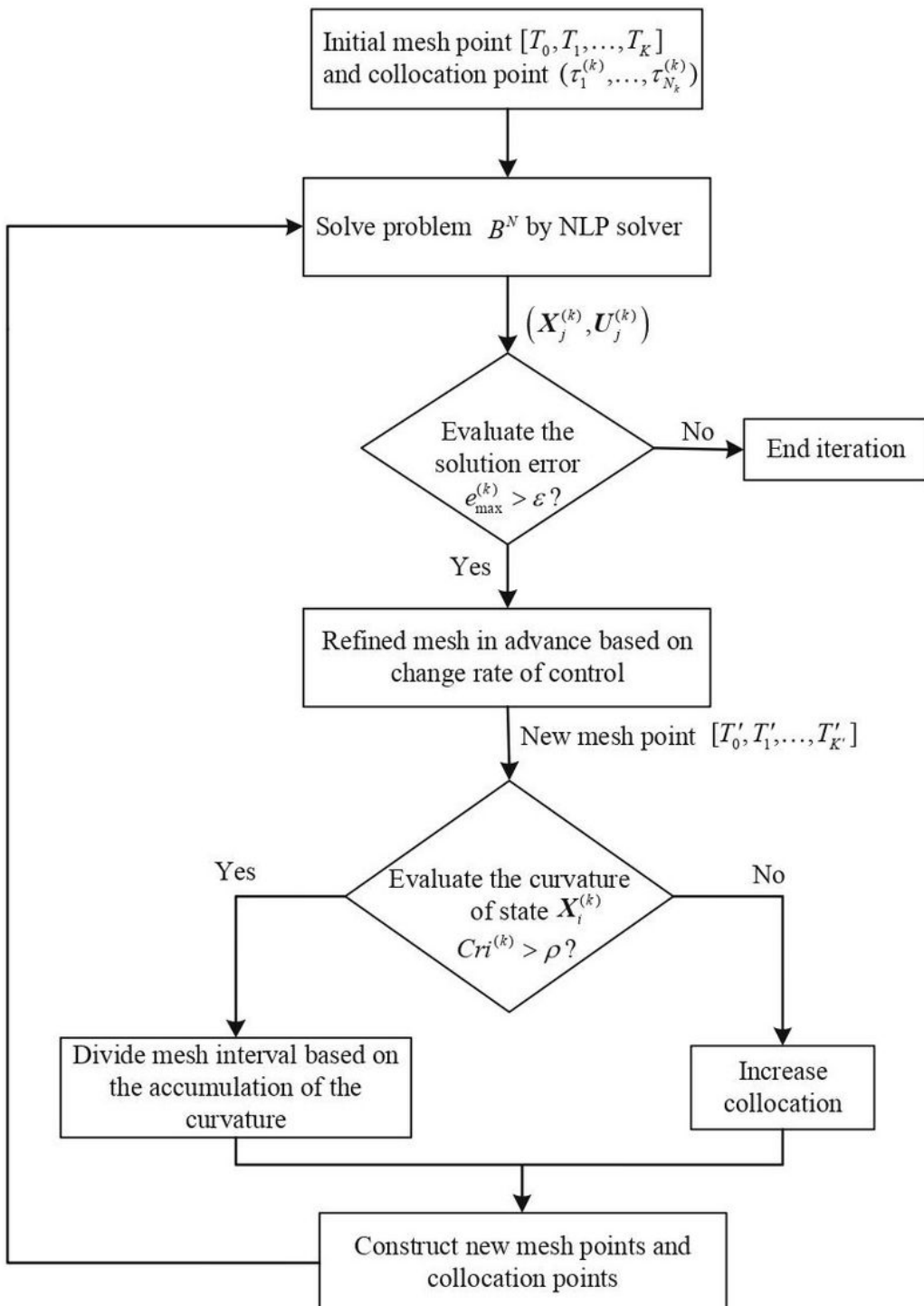
**Figure 2**

The derivative of control.



**Figure 3**

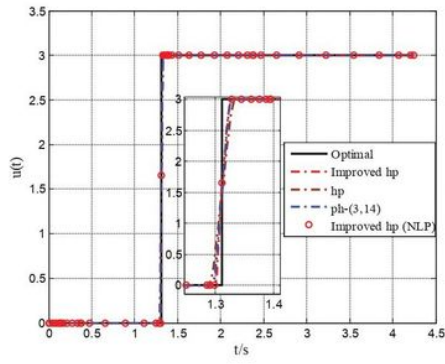
Mesh interval segmentation.



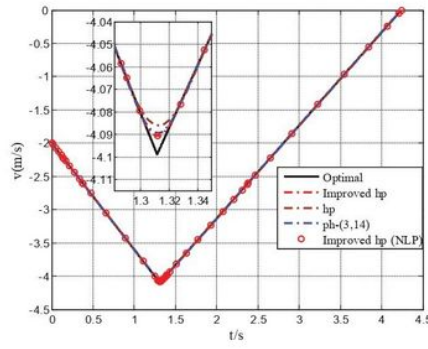
**Figure 4**

Algorithm procedure.

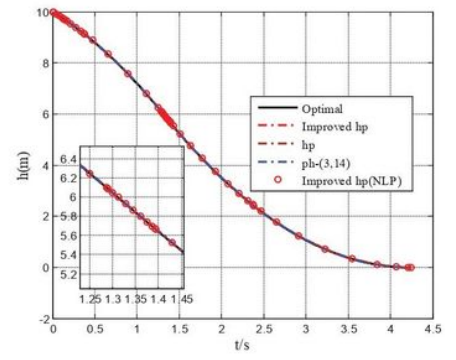




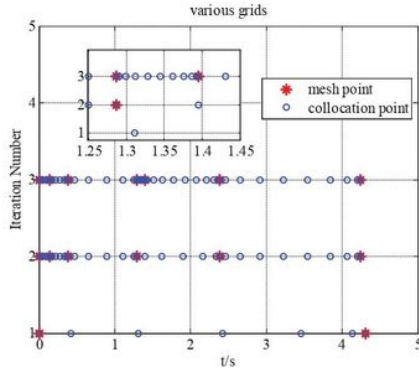
(a) control vs time



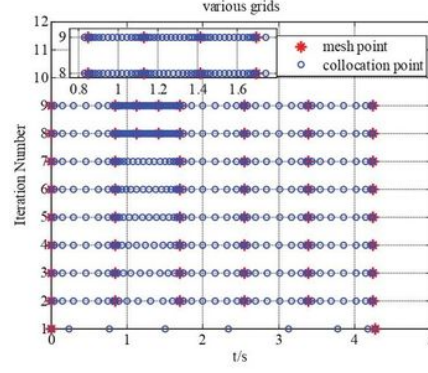
(b) velocity vs time



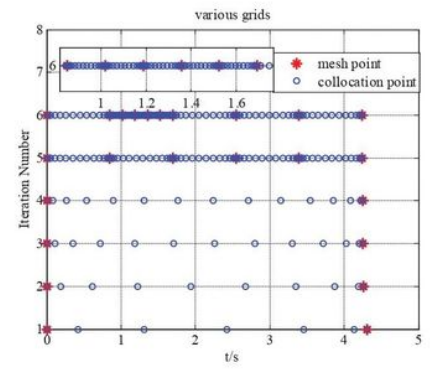
(c) height vs time



(d) various grids of improved hp



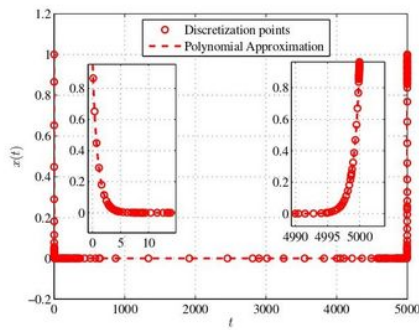
(e) various grids of hp



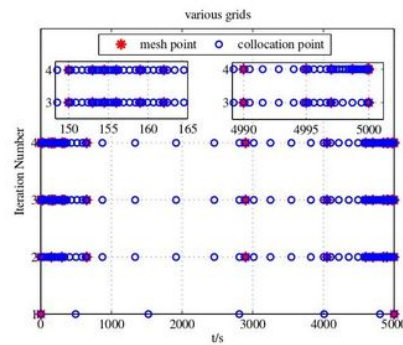
(f) various grids of ph-(3,14)

Figure 5

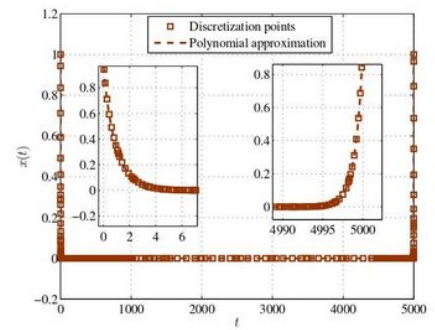
Optimization results of example 1



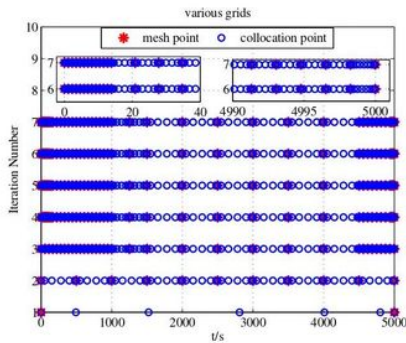
(a)  $x(t)$  of improved hp



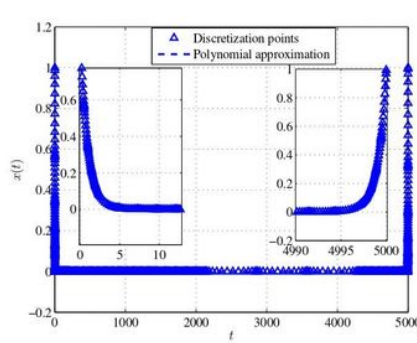
(b) various grids of improved hp



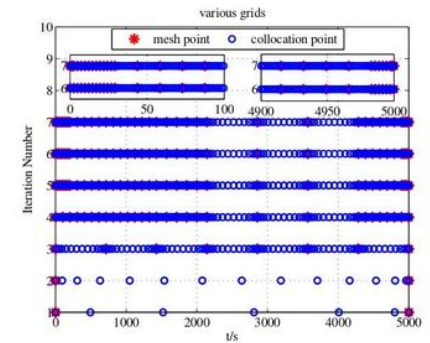
(c)  $x(t)$  of hp



(d) various grids of hp



(e)  $x(t)$  of ph-(3,14)



(f) various grids of ph-(3,14)

Figure 6

Optimization results of example 2

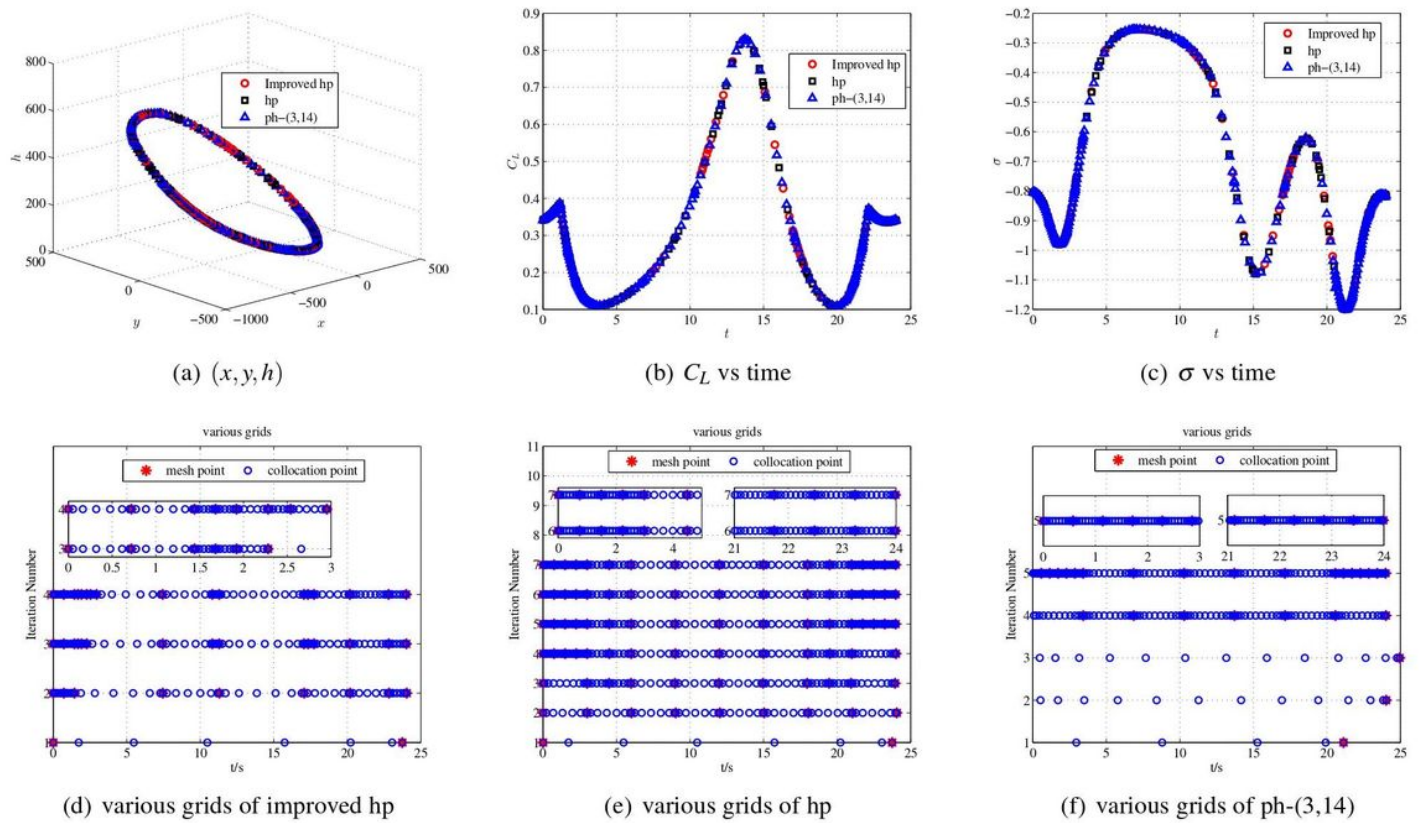


Figure 7

Optimization results of example 3

## Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [NODYSubmissionLaTEX.rar](#)