

Received September 15, 2019, accepted October 22, 2019, date of current version November 15, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2952139

A pk-Adaptive Mesh Refinement for Pseudospectral Method to Solve Optimal Control Problem

JUN HUANG^{1,2}, ZHIGUI LIU^{1,2,3}, ZHIQIN LIU², QINGFENG WANG², AND JIE FU²

¹Institute of Electronic Engineering, China Academy of Engineering Physics, Mianyang 621900, China

²College of Computer Science and Technology, Southwest University of Science and Technology, Mianyang 621010, China

³Information Engineering College, Southwest University of Science and Technology, Mianyang 621010, China

Corresponding author: Zhigui Liu (liuzhigui@swust.edu.cn)

This work was supported in part by the Sichuan Provincial Open Foundation of Civil-Military Integration Research Institute under Grant 2017SCII0219 and Grant 2017SCII0220.

ABSTRACT In this paper, a pk-adaptive mesh refinement of pseudospectral method is proposed for solving optimal control problem by using collocation at Legendre-Gauss-Lobatto (LGL) points, motivated by reducing the redundant collocation points in the state-of-art mesh refinement methods to improve the time efficiency. The proposed method involves three phases, i.e., the determination of the polynomial degree, the determination of increasing intervals or nodes, and the optimization of the locations of segment breaks in each interval. First, determines the polynomial degree by the error estimation between the dynamics and the differentiation approximation of state variables according to the spectral matrix. Second, the maximum allowed polynomial degree in an interval is used to decide whether to segment interval or not. Third, the locations of segment points are obtained as the optimal design parameters of optimal control method. The terminology “pk-adaptive” or “p-then-k adaptive” is used because the polynomial degree is preferentially adaptive variation, then increases the segments by adding the optimal knots in each mesh interval. Finally, the residual of solutions, number of segments, number of nodes, CPU time, convergence of iteration, and parameters of the method have been analyzed in the comparing test to discuss the advantages of pk-adaptive mesh refinement. The discussions performed in two examples and demonstrated that the pk-adaptive method has the ability of optimizing nodes distribution to keep fewer nodes requirement and higher time efficiency than the hp- or ph-based pseudospectral methods while achieving the equivalent accuracy.

INDEX TERMS Optimal control, mesh refinement, pseudospectral, collocation methods, optimal knotting.

I. INTRODUCTION

Pseudospectral methods is widely used in the numerical solution of nonlinear optimal control problem [1], whose examples range from missiles' dive phase trajectory maneuver [2], control of wave energy converters [3], trajectory optimization of boost-glide vehicle [4], trajectory design for lunar landing [5], etc. One of the key points for the wide application of pseudospectral method is the spectral accuracy of exponential convergence in differential approximation theory [6]. The selection of orthogonal basis functions and the orthogonal quadrature rules are two important factors to determine the nodes distribution for differential approximation with few

discrete points [7]. Three orthogonal polynomials, i.e., Legendre [8], Chebyshev [9] and Laguerre [10], are commonly used as the basis functions. And three commonly used orthogonal quadrature rules are Gauss [11], Gauss-Radau [12], and Gauss-Lobatto [13]. By combining the above two factors, a pseudospectral method can be obtained to solve the nonlinear optimal control problems. For example, the Legendre-Gauss-Lobatto (LGL) pseudospectral method [14] combines Gauss-Lobatto quadrature with the Legendre polynomials according to the collocation points, which are known as LGL nodes. There is no definite evidence for the superiority of the selection of orthogonal basis functions between Legendre and Chebyshev, but Laguerre is the only one to be normally discussed for solving the infinite time problems [15]. Fahroo and Ross [16] discussed the application conditions of

The associate editor coordinating the review of this manuscript and approving it for publication was Ton Do ¹.

three orthogonal quadrature rules based on Legendre pseudospectral method and argued that Gauss-Lobatto should be used more in addition to special boundary problems. In contrast, Rao's team used additional and separate treatments for the terminal points with Legendre-Gauss-Radau(LGR) method and demonstrated that the LGR is applicable to both the finite-horizon and infinite-horizon control problems [17], whereby Rao's related research results [18] focused on LGR pseudospectral method.

Both the LGL and LGR pseudospectral methods have been theoretically proved the first-order necessary conditions of optimality through costate mapping to Pontryagin's Maximum Principle, details in [19] and [17] respectively.

Hence, the solution of the optimal control problem can be approximated to a finite degree polynomial, which can be solved by pseudospectral method with enough nodes, and the spectral accuracy of exponential convergence is the same as spectral methods [6]. However, above-mentioned approaches do not adequately address two issues between theory and practice: (a) how to determine the number of nodes and (b) how to deal with solutions that are difficult to approximate to the finite degree polynomial. Mesh refinement sheds a light on solving the above issues, such as spectral algorithm [20] and widely used hp-adaptive method [21].

Mesh refinement (also known as grid refinement [22] in some literatures) techniques are mainly used to obtain the specified solution accuracy in the pseudospectral method of optimal control problems. In the early stage, h method of constant degree polynomial state approximation [23] was adopted to obtain a better accurate solution by increasing the number of trajectory segments. The p method of Gaussian quadrature collocation [7] was proposed to achieve convergence by increasing the degree of the polynomial approximation in the single interval mesh. The limitations of the h method and p method discussed in [21] and [24] are that extremely fine meshes might be required in the h method and an unreasonable large degree polynomial approximation is required in the p method to achieve the expected accuracy tolerance. Therefore, hp [21] method was proposed to allow the variation of the number of mesh intervals and the degree of the approximating polynomial within each mesh interval, which successfully solved many practical problems such as spacecraft attitude maneuver [25], space vehicles trajectory optimization [26], robot path planning [27], etc. Similar to hp method, spectral algorithm proposed by Gong *et al.* [20] before Rao's hp method divided the mesh intervals by placing knots around the control mutation points, and then attempted to increase the number of nodes (relative to the degree of polynomial) at every interval. Recently, Birkhoff interpolation [28] was introduced into the spectral algorithm to improve the poor convergence rate caused by the increase of segments. Patterson *et al.* [29] changed the order from hp to ph, that is, increasing the degree of polynomial before increasing the number of mesh intervals to achieve mesh refinement, so as to improve computational efficiency of pseudospectral method. A mesh size reduction patch [30]

was further developed to reduce the unnecessary points in the iterative steps of ph mesh generation. In summary, both the hp and ph method have been developed to allow variations in the degree of the approximating polynomial in each mesh and the number of intervals in whole mesh. The difference between the hp and ph method lies in the order of varying: varying number of mesh intervals (h) and varying degree of polynomial(p). The hp method can solve the complex optimal control problems in the case of that a global pseudospectral method is computationally intractable, while the ph method has a higher computational efficiency than the hp method in solving the continuous time nonlinear optimal control problems. One of the weaknesses of the ph and hp methods is that the segmentation location cannot be placed accurately, which requires a lot of iterative calculations to approach the segmentation point step by step and the density of discrete points near the segmentation point is increased iteratively until the tolerance of the problem is satisfied. This iteration is particularly time consuming, especially for practical problems in engineering, because each iteration solves an NLP problem which itself contains a large number of numerical iterations.

In order to reduce mesh refinement iteration and improve computational efficiency without losing the accuracy of solution in pseudospectral method, a novel pk-adaptive mesh refinement method based on LGL quadrature collocation is proposed in this study to solve the nonlinear optimal control problems. Here, the word ph is deliberately changed to pk because our scheme attempts to increase the segments by adding optimal knots into the mesh intervals. The optimal knots refer to that the segmentation locations between mesh intervals should be optimized, which is considered as a part of the solution of the NLP problem transformed from optimal control problem. Originally, knotting method [31] was implemented to solve the non-smooth or switch optimal control problem by pre-allocating knots as spectral patches at the switch points. Our method incorporates the idea of knotting into the hp and ph methods, which can be divided into the following three parts. First, as same as the ph method in [29], the error estimation is implemented by comparing the spectral differential value and the value of system dynamic function within a higher order state approximation. Second, the error estimation is used to determine how many numbers of the degree for the polynomial approximation should be increased, or how many optimal knots should be added into a mesh interval. The method in this paper follows the same strategy as in the hp approach [21], that is, increasing collocation points for uniform-type errors and performing segment division for nonuniform-type errors. Third, the uniqueness of our study is that the segments are added by the optimal knots with the differential error nodes as the initial knotting locations, and the final knotting locations are determined by solving the NLP problem which is the same one problem transformed by pseudospectral method itself. In theory, this method will reduce a lot of computation compared with both of the hp and ph methods, because there are a large number of

iterations (solving NLP once is an iteration unit) in the hp and ph methods, but only a small number of optimization design variables are added to determine the locations of segment division for the pk method.

There are four significances of this research. First, the optimal control problem with optimal knots is constructed and converted to NLP problem by discretization of multi-segment LGL pseudospectral method. Second, a new iterative strategy for mesh refinement is devised, in which the polynomial degree of each interval, the number of the intervals in mesh and the segmentation locations between intervals all are allowed to vary. Third, the error estimation and residual vector obtained by difference between system dynamics and spectral differential of higher order approximation polynomials are used to derive exactly the degree of a polynomial in a mesh interval or to determine the numbers and the initial locations of optimal knots. Fourth, two classical examples are presented to illustrate that the pk-adaptive method proposed in this paper produces smaller meshes and has higher computational efficiency with the same specified accuracy tolerance compared with the hp and ph methods.

II. MOTIVATION FOR NEW pk-ADAPTIVE COLLOCATION METHOD

In order to motivate the study of new pk-Adaptive LGL collocation method and to illustrate the fairness of the research, the following first-order differential equation is considered on the interval $\tau \in [-1, 1]$ as same as reference [29]:

$$\frac{dy}{d\tau} = f(\tau) = \begin{cases} 0 & -1 \leq \tau < -0.5 \\ \pi \cos(\pi \tau) & -0.5 \leq \tau \leq +0.5 \\ 0 & -0.5 < \tau \leq +1 \end{cases} \quad (1)$$

where $y(-1) = y_0$. The solutions to the differential equations (1) is given as

$$y(\tau) = \begin{cases} y_0 & -1 \leq \tau < -0.5, \\ y_0 + 1 + \sin(\pi \tau) & -0.5 \leq \tau \leq +0.5, \\ y_0 + 2 & -0.5 < \tau \leq +1. \end{cases} \quad (2)$$

The limitations of p method and h method have been presented, and the advancements of hp method and ph method have been proposed as well for approximating the solution to the differential equation (1) in [21] and [29] respectively. To demonstrate the effectiveness of the pk method in differential approximation, we try to compare the numerical solutions of three mesh refinement strategies: hp, ph, and pk. (1) The core strategy of the hp method is to determine the locations of new segments or increase the number of collocation points according to the type of error distribution. In short, the strategy is to divide the segments for nonuniform-type errors and to increase collocation points for uniform-type errors. (2) The ph method adopts a “p-then-h” strategy where p refinement is exhausted prior to perform any h refinement, and the number of polynomial degree and the number of subintervals can be decided by quantitative calculation formula. (3) The pk

method adopts a “p-then-k” strategy, in which the polynomial degree is changed adaptively and preferentially, and then the segments are increased by adding the optimal knots in each mesh interval.

First of all, to make the study comparable and concise, a convention is made that the three methods in this paper use the unified error form and adopt the pseudospectral method in differential form. The convention before the study is feasible, because the equivalence of differential form and integral form for pseudospectral method have been fully proved in previous studies [32]. As we know, differentiation approximation by pseudospectral matrix D [8] is a key step for solving the optimal control problem by pseudospectral method. Suppose now that it is desired to approximate the differential value of equation (2) using aforementioned mesh refinement strategies (hp, ph, and pk), and the approximation error E can be obtained by comparing with value of equation (1).

$$E = [e^{(1)}, \dots, e^{(H)}]; \quad e^{(i)} = \left| D^{(i)}Y^{(i)} - \frac{(t_f^{(i)} - t_0^{(i)})}{2} f(\tau^{(i)}) \right|. \quad (3)$$

where $i = 1, \dots, H$ represents the sequence number of mesh intervals, and H is the total count of mesh intervals. $\tau^{(i)} = [\tau_1^{(i)}, \tau_2^{(i)}, \dots, \tau_{N^{(i)}+1}^{(i)}]$ is the discrete points of i-th mesh interval at $[t_0^{(i)}, t_f^{(i)}]$ generated by any of the aforementioned strategies. $D^{(i)}$ is the $(N^{(i)} + 1) \times (N^{(i)} + 1)$ LGL differentiation matrix defined on the mesh interval $[t_0^{(i)}, t_f^{(i)}]$. $Y^{(i)}$ can be directly obtained by $y_j^{(i)} = y(\tau_j^{(i)})$. The error convergence and mesh points history are observed through the mesh adaptive iterative procedure of the three strategies. The iterative procedure terminates when all the elements of E are less than the tolerance ε , and we set $\varepsilon = 1 \times 10^{-6}$.

There are three parameters in the hp method denoted as $hp(N_0, L, \rho)$ [21]: N_0 is the initial number of collocation points, then the number of L increases iteratively if needed, and the notion of ρ is a user-defined parameter determining whether adding segments or increasing collocation points. Figure 1 shows the base-10 logarithm of E and mesh point history of iteration for approximating the differential value of equation (2) using $hp(5, 5, 3)$ method. Initial mesh contains 6 collocation points (also, called 6 LGL nodes corresponding to the fifth degree polynomial) in an interval, denoted as “p6h1”. The differential error, drawn as the first one blue circle in the left of figure 1, is approximated on the initial mesh that is plotted as the first rows of blue circles in the right of figure 1. It can be seen that the solving process is convergent through 24 iterations, and the number of nodes is 117 and the number of intervals is 17 in the final mesh when the error satisfies the preset tolerance.

In the ph-adaptive method, the terminology $ph(N_{min}, N_{max})$ refers to the ph refinement method where the polynomial degree can vary between N_{min} and N_{max} . On the same consideration aspects, figure 2 shows the base-10 logarithm of E and mesh point history of iteration for approximating the

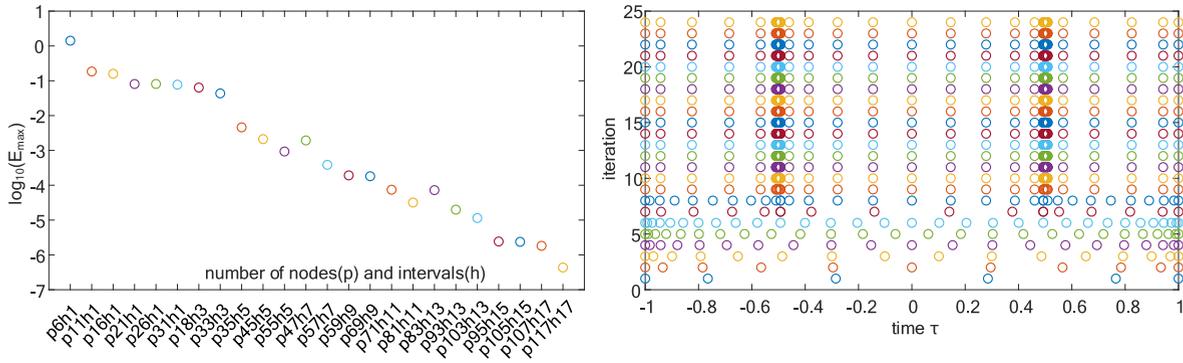


FIGURE 1. Convergence and mesh history using $hp(5, 5, 3)$ method.

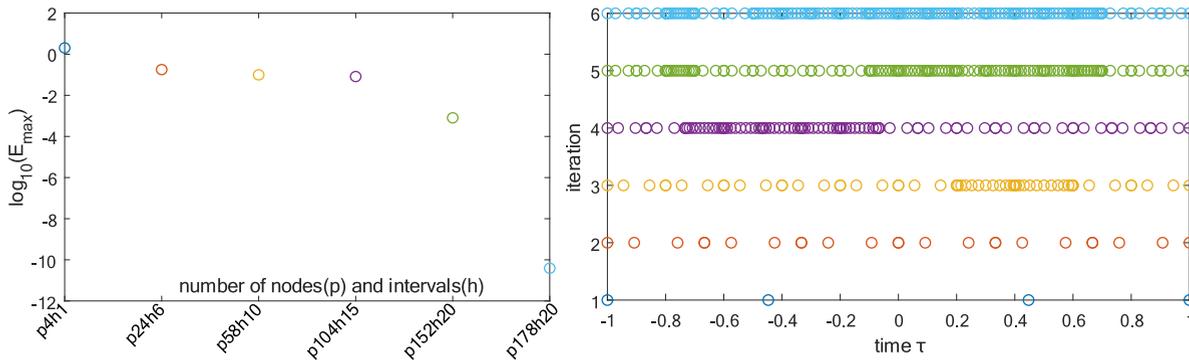


FIGURE 2. Convergence and mesh history using $ph(3, 14)$ method.

differential value of equation (2) using $ph(3, 14)$ method. As can be seen from the figure, the solving process is convergent through only 6 iterations, which is far less than the iterations of $hp(5, 5, 3)$. However, when the error satisfies the preset tolerance, the number of nodes is 178 and the number of intervals is 20.

Both hp and ph methods can converge to the specified precision, but it is not difficult to find a large number of redundant nodes. We assume that these redundant nodes shall be reduced if segment locations could be located more precisely. Therefore, the proposed pk method uses a “p-then-k” strategy where the knots are added by optimal method after p refinement.

There are 3 user-defined parameters for pk method proposed in this paper, which are denoted as $pk(N_{min}, N_{max}, \rho)$. The details of the parameters and pk strategy will be discussed in Section V later. It is seen that the solving process is convergent through only 6 iterations, and the number of nodes is 52 and the number of intervals is 6 when the error satisfies the preset tolerance. Both the number of nodes and intervals are significantly smaller than the previous two methods, and convergence requirements are the same.

The pk method may cost more time for differential approximation in an iteration because the optimal step is added. However, it only adds several number of optimal design parameters, the number of which is equal to the number of segments break points, and is far less than the number of parameters added by increasing nodes in pseudo

spectral method for the optimal control problem. It is possible to improve the total computational efficiency by using a pk -adaptive method.

III. OPTIMAL CONTROL PROBLEM WITH OPTIMAL KNOTS

Without the loss of generality, the following general optimal control problem in Bolza form is considered to minimize the cost function

$$J[x(t), u(t), t_f] = \Phi(x(t_0), t_0, x(t_f), t_f) + \int_{t_0}^{t_f} g(x(t), u(t), t)dt \quad (4)$$

subject to the dynamic constraints

$$\frac{dx}{dt} = f(x(t), u(t), t) \quad (5)$$

the inequality path constraints

$$c(x(t), u(t), t) \leq 0 \quad (6)$$

and the boundary conditions

$$b(x(t_0), t_0, x(t_f), t_f) = 0 \quad (7)$$

where $x(t)$ is the state, $u(t)$ is the control, and $t \in [t_0, t_f]$ is time.

Suppose the time interval $t \in [t_0, t_f]$ is divided into a mesh, which is composed of K mesh intervals: $I^{(k)} = [T_{k-1}, T_k]$, $k = 1, \dots, K$. And $T_{break} = [T_1, \dots, T_{K-1}]$ is an array of

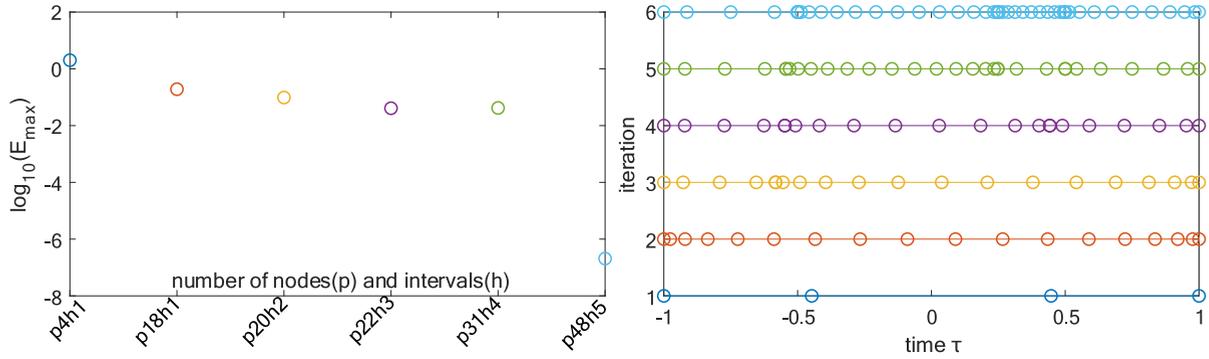


FIGURE 3. Convergence and mesh history using $pk(3, 14, 2)$ method.

the segment break points, called knots. The mesh intervals I_k have the properties of $\bigcup_0^K I^{(k)} = [t_0, t_f]$ and $\bigcap_0^K I^{(k)} = T_{break}$, while the mesh points have the properties of $t_0 = T_0 < T_1 < \dots < T_K = t_f$. The LGL node points lie in the domain $[-1, 1]$ in the Legendre approximation. So the time intervals $t^{(k)} \in [t_0^{(k)}, t_f^{(k)}] = I^{(k)}$ are related as the following affine transformation:

$$t^{(k)} = (t_f^{(k)} - t_0^{(k)})\tau^{(k)}/2 + (t_f^{(k)} + t_0^{(k)})/2. \quad (8)$$

Let $x^{(k)}(\tau)$ and $u^{(k)}(\tau)$ be the state and control in $I^{(k)}$, respectively. The Bolza optimal control problem of (4)-(7) can be then rewritten as the following multi-intervals problem with the location of knots T_{break} as the optimal design parameters to minimize the cost functional

$$J[x^{(1)}(\tau^{(1)}) \dots x^{(K)}(\tau^{(K)}); u^{(1)}(\tau^{(1)}) \dots u^{(K)}(\tau^{(K)}); T_{break}; t_f] = \Phi(x^{(1)}(-1), t_0^{(1)}, x^{(K)}(1), t_f^{(K)}) + \sum_{k=1}^K \left[\frac{t_f^{(k)} - t_0^{(k)}}{2} \int_{-1}^1 g(x^{(k)}(\tau), u^{(k)}(\tau), \tau) d\tau \right] \quad (9)$$

subject to the dynamic constraints

$$\frac{dx^{(k)}}{d\tau} = \frac{t_f^{(k)} - t_0^{(k)}}{2} f(x^{(k)}(t), u^{(k)}(t), t), \quad k = 1 \dots K \quad (10)$$

the path constraints

$$c(x^{(k)}(\tau^{(k)}), u^{(k)}(\tau^{(k)}), t^{(k)}) \leq 0 \quad (11)$$

and the boundary conditions

$$b(x^{(1)}(-1), t_0^{(1)}, x^{(K)}(1), t_f^{(K)}) = 0. \quad (12)$$

Since the state of each interior mesh point must be continuous, it requires that the condition $x^{(k-1)}(1) = x^{(k)}(-1)$ and the optional condition $u^{(k-1)}(1) = u^{(k)}(-1), k = 1 \dots K$ should be satisfied at the knots ($T_{break} = [T_1, \dots, T_{K-1}]$).

IV. LEGENDRE-GAUSS-LOBATTO COLLOCATION METHOD WITH MULTI-SEGMENT

The optimal control problem with optimal knots in section III is discretized by using collocation at Legendre-Gauss-Lobatto(LGL) [16] points. In the LGL collocation approximation of (9)-(12), the state of any one interval $I_k, k \in [1, \dots, K]$ is approximated as shown in equation (13) by using the discrete state variables at the LGL points $\tau^{(k)} = \{\tau_i^{(k)}\}, i = 1, \dots, N^{(k)} + 1$. The $\tau_i^{(k)}$ are given as $\tau_1^{(k)} = -1, \tau_{N+1}^{(k)} = 1$, and for $1 < i < N^{(k)} + 1, \tau_i^{(k)}$ are the zeros of \dot{L}_N which is the derivative of the Legendre polynomial L_N .

$$x^{(k)}(\tau) \approx x^{N^{(k)}}(\tau) = \sum_{i=1}^{N^{(k)}+1} x_i^{(k)} \varphi_i^{(k)}(\tau), \quad \varphi_i^{(k)}(\tau) = \prod_{\substack{j=1 \\ j \neq i}}^{N^{(k)}+1} \frac{\tau - \tau_j^{(k)}}{\tau_j^{(k)} - \tau_i^{(k)}} \quad (13)$$

where $x^{N^{(k)}}(\tau)$ is interpolation of $x^{N^{(k)}} = \{x_i^{(k)}\}, x_i^{(k)} = x^{(k)}(\tau_i)$ is the value of state at the collocation point τ_i , and $\varphi_i^{(k)}(\tau)$ is a basis of Lagrange polynomial. The differential of state can be expressed as a matrix multiplication form in terms of $x_i^{(k)}$:

$$\dot{x}^{(k)}(\tau_i) = \sum_{j=1}^{N^{(k)}} D_{ij}^{(k)} x_j^{(k)}, \quad (14)$$

where $D_{ij}^{(k)}$ are entries of the $(N^{(k)} + 1) \times (N^{(k)} + 1)$ differentiation matrix $D^{(k)}$ associated with $I_k, k \in [1, \dots, K]$. Next the integral form in Eq.(9) is discretized as Eq.(15) by using the Gauss-Lobatto integration rule.

$$\int_{-1}^1 g(x^{(k)}(\tau), u^{(k)}(\tau), \tau) d\tau = \sum_{i=1}^{N^{(k)}+1} [g(x_i^{(k)}, u_i^{(k)}, \tau_i^{(k)}) w_i] \quad (15)$$

where w_i are the weights for approximate integral on discretized LGL nodes.

In summary, the optimal control problem (9)-(12) is approximated by the following nonlinear optimization problem (NLP): Find state $x^{N^{(k)}}$, control $u^{N^{(k)}}$, location of knots

T_{break} and possibly final time t_f to minimize

$$J[x^{N^{(1)}} \dots x^{N^{(K)}}; u^{N^{(1)}} \dots u^{N^{(K)}}; T_{break}; t_f] = \Phi(x_1^{(1)}, t_0^{(1)}, x_{N^{(k)+1}^{(1)}}, t_f^{(K)}) + \sum_{k=1}^K \left\{ \frac{t_f^{(k)} - t_0^{(k)}}{2} \sum_{i=1}^{N^{(k)+1}} [g(x_i^{(k)}, u_i^{(k)}, \tau_i^{(k)}) w_i] \right\} \quad (16)$$

subject to

$$\sum_{j=1}^{N^{(k)}} D_{ij}^{(k)} x_i^{(k)} - \frac{t_f^{(k)} - t_0^{(k)}}{2} f(x_i^{(k)}, u_i^{(k)}, t_i^{(k)}) = 0, \quad k = 1 \dots K, i = 1 \dots N^{(k)} \quad (17)$$

$$c(x_i^{(k)}, u_i^{(k)}, t_i^{(k)}) \leq 0, i = 1 \dots N^{(k)} + 1 \quad (18)$$

$$b(x_1^{(1)}, t_1^{(1)}, x_{N^{(k)+1}^{(K)}}, t_{N^{(k)+1}^{(K)}}) = 0 \quad (19)$$

$$x_{N^{(k-1)+1}^{(k-1)}} = x_1^{(k)}, u_{N^{(k-1)+1}^{(k-1)}} = u_1^{(k)}. \quad (20)$$

V. PK-ADAPTIVE MESH REFINEMENT METHOD

A pk-adaptive mesh refinement method is presented for solving optimal control problem by using collocation at Legendre-Gauss-Lobatto (LGL) points described in Section IV. The ‘‘pk-adaptive’’ or ‘‘p-then-k adaptive’’ is used because the polynomial degree adaptive changed preferentially then increased the segments by adding the optimal knots in each mesh interval.

A. ERROR ESTIMATION OF EACH MESH INTERVAL

The Error estimation is based on how closely the dynamic constraints are satisfied at the new collocation points, $\{\hat{\tau}_i^k, i = 1, \dots, N^{(k)} + 2\}$, which are the new LGL points by increasing collocation number. Assume $x^{N^{(k)}}, u^{N^{(k)}}, t_0^{(k)}, t_f^{(k)}$ are the solutions of NLP described by Eqs.(16)-(20) on a mesh $I_k, k \in [1, \dots, K]$, corresponded with $N^{(k)} + 1$ LGL points. Then the error is estimated on the state at a set of $N^{(k)} + 2$ LGL points ($\hat{\tau}^{(k)} = [\hat{\tau}_1^{(k)}, \dots, \hat{\tau}_{N^{(k)+2}^{(k)}}]$), where $\hat{\tau}_1^{(k)} = -1, \hat{\tau}_{N^{(k)+2}^{(k)}} = 1$, and for $2 \leq i \leq N^{(k)} + 1, \tau_i^{(k)}$ are the zeros of $\hat{L}_{N^{(k)+1}^{(k)}}$. The Error of I_k is obtained by Eq.(21), which is similar with Eq.(3) in Section II.

$$E^{(k)} = \left| \hat{D}^{(k)} \hat{x}^{N^{(k)}} - \frac{(t_f^{(k)} - t_0^{(k)})}{2} f(\hat{x}^{N^{(k)}}, \hat{u}^{N^{(k)}}, \hat{\tau}^{(k)}) \right| \quad (21)$$

where $\hat{x}^{N^{(k)}} = \{\hat{x}_i^{(k)} | \hat{x}_i^{(k)} \approx x^{N^{(k)}}(\hat{\tau}_i^{(k)})\}, \hat{u}^{N^{(k)}} = \{\hat{u}_i^{(k)} | \hat{u}_i^{(k)} = u^{N^{(k)}}(\hat{\tau}_i^{(k)})\}$ can be calculated by interpolation from $x^{N^{(k)}}, u^{N^{(k)}}$ at the points $\tau_i^{(k)}$ using Eq.(13), and $\hat{\tau}_i^{(k)}$ can be obtained directly by Eq.(8).

The $E^{(k)}$ is the $n_x \times (N^{(k)} + 2)$ matrices that assume the number of states is n_x . The maximum error in I_k is then defined as

$$e_{max}^{(k)} = \max_{i \in [1 \dots n_x], j \in [1 \dots N^{(k)+2}]} (E^{(k)}) \quad (22)$$

Let the residual vector $r^{(k)}$ be the elements of the column of the residual matrix $E^{(k)}$ that contains the largest value of $E^{(k)}$.

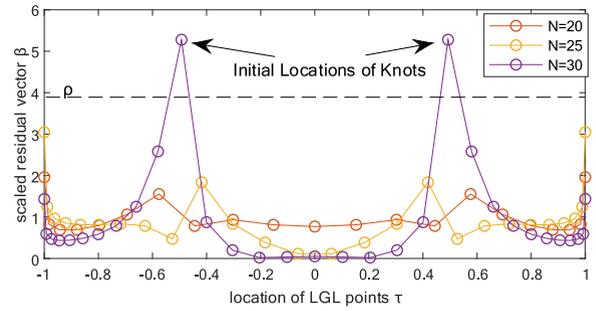


FIGURE 4. Initial locations of knots obtained by increasing polynomial degree.

Then the $r^{(k)}$ can be written in component form as

$$r^{(k)} = \max_{l \in [1 \dots n_x]} (E^{(k)}) = [e(\hat{\tau}_1^{(k)}) \dots e(\hat{\tau}_{N^{(k)+2}^{(k)}})]^T. \quad (23)$$

The scaled residual vector is then defined as

$$\beta^{(k)} = [e(\hat{\tau}_1^{(k)})/\bar{e} \dots e(\hat{\tau}_{N^{(k)+2}^{(k)}})/\bar{e}]^T, \quad (24)$$

where \bar{e} is mean value of the vector $r^{(k)}$.

B. POLYNOMIAL DEGREE IN EACH MESH INTERVAL

The method of estimating the required polynomial degree within a mesh interval has been proposed based on the convergence theory. Suppose that interval I_k employs $N^{(k)}$ collocation points and has maximum error $e_{max}^{(k)}$ which is larger than the desired error tolerance ϵ . To reach the desired error tolerance, the error reduction is achieved by increasing $N^{(k)}$ by $P^{(k)}$, and $P^{(k)}$ is chosen as an integer not less than 3 for the robustness of the method. The value of $P^{(k)}$ are expressed as follow:

$$P^{(k)} = \max(\left\lceil \log_{N^{(k)}} \left(\frac{e_{max}^{(k)}}{\epsilon} \right) \right\rceil, 3). \quad (25)$$

C. NUMBER AND INITIAL LOCATIONS OF KNOTS

The scaled residual vector $\beta^{(k)}$ will show as ‘‘nonuniform-type’’ if the $e_{max}^{(k)}$ is still larger than the desired error tolerance ϵ after increasing polynomial degree. Figure 4 shows that the scaled residual vector was iteratively obtained by comparing the spectral differential of equation (2) with (1). The ‘‘nonuniform-type’’ errors characteristic is presented with the increasing of polynomial degree. According to (2), we know that the locations of $\tau = -0.5$ and $\tau = 0.5$ are the best segment breaks, and there are two LGL points in the elements of $\beta^{(k)}$ that exceed near the best segment breaks. The points where the entries of vector $\beta^{(k)}$ exceed ρ are used as the initial locations of knots, because it doesn’t have to be LGL points that happen to coincide with the best segment breaks while the best segment breaks must be near the ‘‘nonuniform-type’’ errors points. Three special treatments were considered to enhance the robustness of the ρ value setting. First, when the vector $\beta^{(k)}$ contains adjacent entries greater than ρ , the initial locations of knots are only the locations of the largest element

of the adjacent entries. Second, when there is no element greater than ρ , the only initial location of knot is placed at the location of the largest element of the $\beta^{(k)}$ after the loss of convergence gain by increasing polynomial degree. Third, the elements of $\tau = -1$ and $\tau = 1$ should be excluded when comparing elements of vector $\beta^{(k)}$ to ρ .

In short, the initial locations of knots are obtained as LGL points on the relation between $\beta^{(k)}$ and ρ based on the above mentioned rules, then the simplified expression is denoted as follows:

$$\tau_{knots}^{(k)} = \begin{cases} \{\tau_i^{(k)} | adjacentmax(\beta_i^{(k)} > \rho)\}, & i \in [2, N] \\ \{\tau_i^{(k)} | max(\beta_i^{(k)})\}, & if \{\beta_i^{(k)} > \rho\} is empty \end{cases} \quad (26)$$

To solve the NLP problem expressed as equation (16), the initial locations of knots should be transformed into the time-domain of original optimal control problem. Therefore, the initial locations of knots for NLP problem can be expressed as $T_{knots}^{(k)}$ in interval $I^{(k)}$ and can be achieved by combining equation (26) with (8). The number of knots is denoted as $N_{knots}^{(k)}$, which is the size of $T_{knots}^{(k)}$.

D. “P-THEN-K” MESH REFINEMENT STRATEGY

If $e_{max}^{(k)} > \varepsilon$, the method proposed in this paper gives priority to increase polynomial degree in an interval $I^{(k)}$ until $\hat{N}^{(k)} > N_{max}$ (that is, $\hat{N}^{(k)}$ exceeds the maximum allowable polynomial degree specified by the user). The method of adding optimal knots should be proceeded on interval $I^{(k)}$ if $e_{max}^{(k)}$ still has not reached the specified error tolerance ε by increasing polynomial degree. A summary of our adaptive mesh refinement algorithm in pseudospectral appears as **Algorithm 1**.

VI. EXAMPLES

In this Section, two examples from the open literatures are applied to demonstrate the advantages of the pk-adaptive Legendre-Gauss-Lobatto(LGL) method described in Section V. The first example is the optimal control problem of soft lunar landing, which has been used to verify the benefits of hp mesh refinement in [21].The second example is the hyper-sensitive optimal control problem, which has been used to verify the benefits of both hp and ph refinement in [29]. Three mesh refinement methods (hp, ph and pk) are implemented by ourselves on Matlab platform, because there are many special procedures in different original references, including NLP solver IPOPT or SNOPT, initial guess strategies, parallel sparse Matrix solver, error estimation and collocation points.

In order to test the availability of pk mesh refinement, all codes including the error estimation, the LGL methods, and the NLP solver are remain unchanged except for the mesh refinement part, as shown in Figure 5. For simplicity, the fmincon function is used to solve NLP, which is a built-in function of Matlab.

Algorithm 1 “p-then-k” Mesh Refinement of pseudospectral.

Step1: Initialization. set $K = 1, T_{break} = NULL$; put interval $I^{(1)} = [t_0^{(1)}, t_f^{(1)}]$ into initial mesh $M_0 = [I^{(1)}]$.

Step2: Time transformation. Adjust all intervals lie in the $[-1, 1]$. Construct optimal control problem J which contains the knots T_{break} formed as equation (9) in Section III, that is $J[x^{N(1)}(\tau^{(1)}) \dots x^{N(K)}(\tau^{(K)}); u^{(1)}(\tau^{(1)}) \dots u^{(K)}(\tau^{(K)}); T_{break}; t_f]$.

Step3: Construct NLP problem J^N using LGL pseudospectral method formed as equation (16) in Section IV, Then J^N is expressed as $J[x^{N(1)} \dots x^{N(K)}; u^{N(1)} \dots u^{N(K)}; T_{break}; t_f]$.

Step4: Solve J^N using Matlab NLP solver named fmincon. Solutions of J^N include $x^{N(1)} \dots x^{N(K)}, u^{N(1)} \dots u^{N(K)}, T_{break}, t_f$.

Step5: Calculate $\hat{x}^{N(k)}, \hat{u}^{N(k)}$ by interpolation on the $x^{N(k)}, u^{N(k)}$; then calculate scaled error $e_{max}^k, k = 1 \dots K$ using equation (21)-(22) in Section V-A.

Step6: If $e_{max}^k \leq \varepsilon$ is for all $k = 1 \dots K$, then quit; Otherwise, create new empty mesh $M_{new} = []$ and empty array of knots $T_{b,new} = []$, then proceed to Step7.

Step7: Mesh refinement. Iterative over $I^{(k)}$ for $k = 1 \dots K$, and the codes are as follows.

```

for  $k = 1 \rightarrow K$  do
  if  $e_{max}^k \leq \varepsilon$  then
    Append  $I^{(k)}$  into  $M_{new}$ ;
    Append  $T_{break}(k)$  into  $T_{b,new}$ .
  else
    Calculate  $P^{(k)}$  using equation (25);
    Estimate degree of  $I^{(k)}$  as  $\hat{N}^{(k)} = N^{(k)} + P^{(k)}$ ;
    if  $N^{(k)} \leq N_{max}$  then
      Create an interval  $\hat{I}^{(k)}$  with  $\hat{N}^{(k)}$  LGL points;
      Append  $\hat{I}^{(k)}$  into  $M_{new}$ ;
      Append  $T_{break}(k)$  into  $T_{b,new}$ .
    else
      Get scaled residual vector  $\beta^{(k)}$  by Eq. (24);
      Get  $\tau_{knots}^{(k)}, T_{knots}^{(k)}, N_{knots}^{(k)}$  using Eq. (26);
      Add knots that set  $K = K + N_{knots}^{(k)}$ ;
      Create  $N_{knots}^{(k)}$  intervals with  $N_{min}$  LGL points:
       $M^{(k)} = [\hat{I}^{(k,1)} \dots \hat{I}^{(k,N_{knots}^{(k)})}]$ ;
      Append  $M^{(k)}$  into  $M_{new}$ ;
      Append  $T_{knots}^{(k)}$  and  $T_{break}(k)$  into  $T_{b,new}$ .
    end if
  end if
end for

```

Step8: Rebuild NLP problem J^N on new mesh M_{new} with new knots $T_{break} = T_{b,new}$, and get initial guess solution for next NLP of Step 3 by using interpolation of current result. Return to Step 3.

The terminology $hp(N_0, L, \rho)$ refers to the hp-adaptive method where the initial number of collocation points N_0

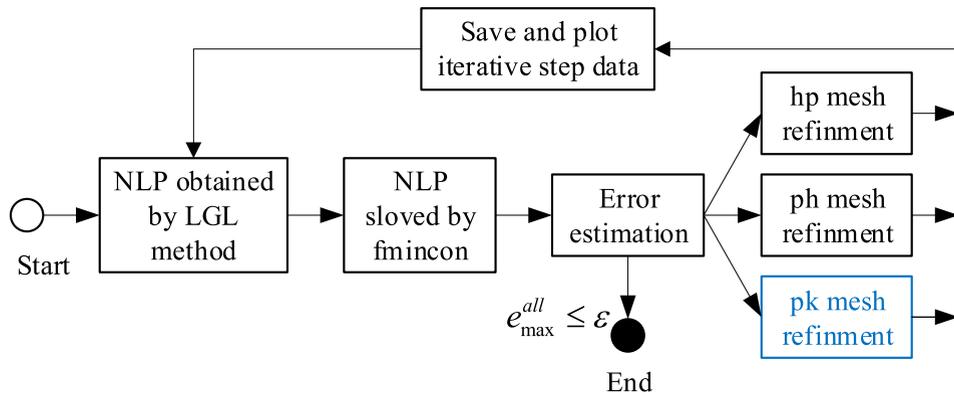


FIGURE 5. Comparing test design for verifying the benefits of pk mesh refinement.

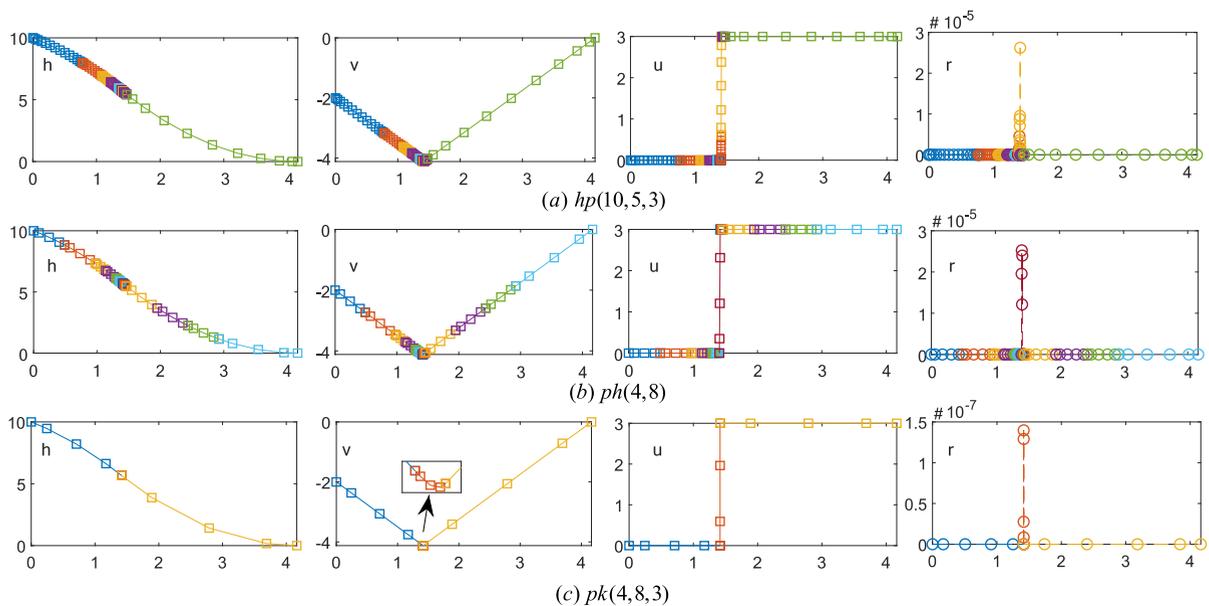


FIGURE 6. The state, control and residual vector on the final mesh grid of Moon-lander problem.

in the segment is increased by the user-specified amount L and segment division is determined by the parameter ρ . The terminology $ph(N_{min}, N_{max})$ refers to the ph-adaptive method where the polynomial degree can vary between N_{min} and N_{max} . The terminology $pk(N_{min}, N_{max}, \rho)$ refers to the pk-adaptive method where the polynomial degree can vary between N_{min} and N_{max} and the number and initial locations of knots are determined by parameter ρ as described in Section V.

A. EXAMPLE 1: BANG-BANG CONTROL PROBLEM OF MOON-LANDER

Consider the following a soft lunar landing which is a classic bang-bang optimal control problem. Minimize

$$J = \int_{t_0}^{t_f} u dt \tag{27}$$

subject to

$$\dot{h} = v, \quad \dot{v} = -g + u \tag{28}$$

the boundary conditions

$$h(0) = 10, \quad h(t_f) = 0, \quad v(0) = -2, \quad v(t_f) = 0 \tag{29}$$

and the control constraint

$$0 \leq u \leq 3 \tag{30}$$

where $g = 1.5$, and t_f is free.

As mentioned above, the three strategies (hp, ph and pk) combined into the same one LGL pseudospectral method were used to solve the moon-lander problem. Firstly, the parameters of hp and ph are set as $hp(10, 5, 3)$ and $ph(4, 8)$ respectively. Then the parameters of ph are set as follows: the N_{min} and N_{max} are the same with ph and the parameter ρ of pk is the same as hp, that is, $pk(4, 8, 3)$.

Figure 6 shows the last iterative results of Moon-lander problem solved by LGL pseudo spectral method with three mesh refinement strategies for $\epsilon = 10^{-4}$. In the figures of this section, a mesh interval is represented by the same color, the color squares represent the location of the nodes, and the

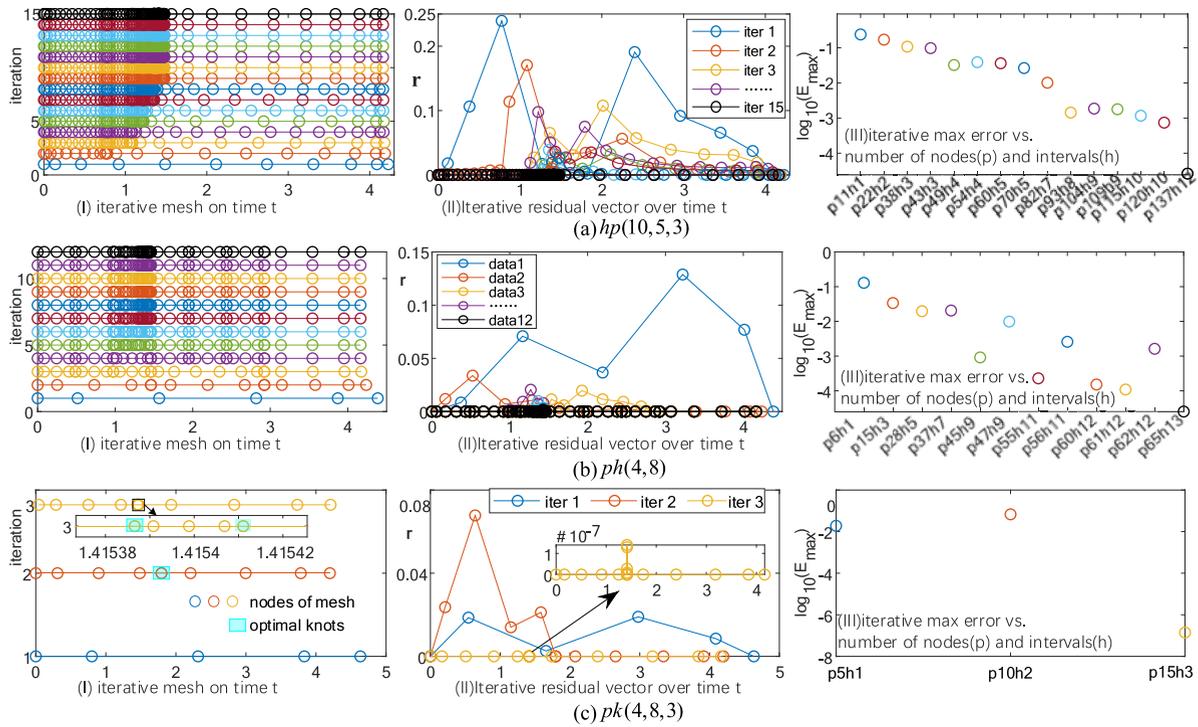


FIGURE 7. Convergence with mesh refinement of Moon-lander.

knots are the nodes connecting two kinds of color intervals. Intuitively, these three methods progress to a final mesh grid, so that the collocation points are more densely located near the control discontinuity. When using the method of $hp(10, 5, 3)$, it is difficult to find the location of the control discontinuity and iteratively generate a number of redundant segments clustered around discontinuities. When using the method of $ph(4, 8)$, the total number of collocation points is much smaller than $hp(10, 5, 3)$ because of that the polynomial degree and the number of segments are calculated by magnitude of error, but the redundant intervals are generated by trying add segments in an interval to reduce the residual. When using the method of $pk(4, 8, 3)$, segment breaks occur at $t=(1.41538658078292, 1.41541106102890)$, which is a small value range containing the location of the control discontinuity by approach of optimal knots, and $pk(4, 8, 3)$ method achieves better results with lower residual and fewer number of grid nodes significantly.

Next the specificity and advantages of pk methods will be discussed. The relation between convergence and mesh refining history has been presented in figure 7. There are three type of plots given as follow: (I) The iterative discrete mesh which includes multi-intervals constructed by LGL nodes, generated by mesh refinement on the time domain of problem. (II)The residual vector which is calculated by equation (23) based on each solution responds iterative mesh. (III)The iterative max error is calculated by equation (22), that a color point represents to the max error of a solution solved corresponding on an iterative mesh. The corresponding relations

of the three plots are presented by the color, where each color represents one iteration. First, the convergence progress is achieved by $hp(10, 5, 3)$ as shown in figure 7(a). The initial mesh contains 1 interval with 11 nodes and is plotted as the first row of blue circles in (I). The corresponding residual vector is plotted as blue circles in (II) and max error is denoted by “p11h1” with the blue circle in (III). After 15 iterations of hp mesh refinement, the maximum error of the solution converges to the solution, which is less than the tolerance ϵ . The last iterative mesh contains 12 intervals with 137 nodes as the black circle in the three plots. Second, figure 7(b) shows that the solution is solved by $ph(4, 8)$ after 12 times of iterations which starts from the “p6h1” and terminals to “p65h13”. An interesting phenomenon can be found that sometimes the max error of ph will raise when the nodes in the new mesh increase, such as the 5-th iteration (“p45h9”) to the 6-th iteration (“p47h9”). This is because the Gibbs problem occurs when nodes increased into the interval containing the switch points. And the error will reduce when adding more segments in the interval as shown in the plot (III) of figure 7(b). Third, The fast convergence process of $pk(4, 8, 3)$ is shown in figure 7(c). After only 3 iterations, only 15 nodes and 3 intervals (“p15h3”) use the max error to meet the tolerance. The residual vector reduces to $1.397e-7$ when 2 knots are placed right at the locations $t=(1.41538658078292, 1.41541106102890)$ that are close to the switch point ($t^*=1.4514$). The most important characteristic of the pk method, shown in (I) of figure 7(c), is that the optimal knots are variable in each iteration of the NLP solver.

TABLE 1. Accuracy and speed in the three mesh refinement methods for moon-lander problem.

Mesh refinement method	CPU time(s)	Collocation points	Segments	No. of grids	e_{max}
$hp(10, 5, 3)$	182.6094	137	12	15	2.624e-5
$hp(10, 10, 3)$	70.0313	119	10	11	5.466e-5
$hp(10, 15, 3)$	618.4688	198	14	17	1.551e-5
$hp(5, 10, 3)$	126.3906	81	12	21	6.207e-5
$ph(4, 8)$	45.5625	65	13	12	2.525e-5
$ph(4, 12)$	30.2188	63	13	11	9.331e-5
$ph(4, 16)$	58.2500	87	17	15	3.991e-5
$ph(5, 8)$	38.3281	83	14	8	9.011e-5
$ph(6, 12)$	59.3281	96	13	13	7.928e-5
$pk(4, 6, 3)$	3.5938	15=3*(4+1)	3	3	1.397e-7
$pk(4, 8, 3)$	3.6563	15=3*(4+1)	3	3	1.397e-7
$pk(4, 12, 3)$	8.7188	15=3*(4+1)	3	5	3.972e-7
$pk(4, 16, 3)$	10.2500	15=3*(4+1)	3	7	1.04e-6
$pk(5, 8, 3)$	3.9219	18=3*(5+1)	3	3	4.329e-6
$pk(6, 8, 3)$	11.8906	21=3*(6+1)	3	3	1.273e-5

In addition, different parameters for the mesh refinement, the accuracy and speed may appear different. Table 1 shows the performance of the pk-adaptive method solving Moon-lander problem in comparison with the other two mesh refinement methods. We try to vary the initial number of N_0 and iteratively increases the amount of L for hp method, so that $hp(10, 10, 3)$ can get better performance than others in increasing N_0 or L . There are 10 intervals and 119 collocation points in the last iterative mesh of $hp(10, 10, 3)$ achieved by 11 iterations with 70.313 seconds. Then, we vary the N_{min} and N_{max} for ph method, so that $ph(4, 12)$ gets better performance than others. After 11 iterations with 30.2188 seconds, the last iterative mesh of $ph(4, 12)$ contains 13 intervals and 63 collocation points. In the same way, we vary the N_{min} and N_{max} for pk method, so that the $pk(4, 6, 3)$ get better performance than others. The 3 iterations take only 3.5938 seconds, and the third iterative meshes only contain 3 intervals with a total of 15 collocation points. Most notably, a pattern was discovered in the pk method that the last iterative meshes always keep 3 intervals and totally contain $3 \times (N_{min} + 1)$ collocation points. The $(N_{min} + 1)$ is the minimum collocation points in an interval. The moon-lander problem only contains a switch point, which requires three intervals at least for infilling the mesh grid near the switch point. In other words, the best mesh can be obtained by the pk method for the moon-lander problem.

B. EXAMPLE 2: HYPER-SENSITIVE PROBLEM

Consider the following hyper-sensitive optimal control problem. Minimize

$$J = \frac{1}{2} \int_0^{t_f} (x^2 + u^2) dt \tag{31}$$

subject to

$$\dot{x} = -x^3 + u \tag{32}$$

the boundary conditions

$$x(0) = 1.5, \quad x(t_f) = 1, \tag{33}$$

where t_f is user-specified and fixed.

It is known that the solution to the hyper-sensitive problem is a three segments structure of “take-off”, “cruise” and “landing”. The “cruise” segment of the solution is constant, and the interesting behavior occurs near the “take-off” and “landing” segments. Furthermore, the cruise segment takes an increasing proportion in the total solution time with the increasing value of t_f , while exponential decay and rapid growth occur respectively in the “take-off” and “landing” segments. According to the above characteristics of the solution, the distribution of the collocation points is expected to be a small number of points in cruise, and most of them should be distributed in “take-off” and “landing” segments.

Firstly, the solutions of hyper-sensitive problem are solved by using LGL pseudospectral within three strategies (hp, ph and pk) respectively. Figure 8 shows the last iterative results of Hyper-Sensitive problem for $t_f = 40$ and $\varepsilon = 10^{-3}$. As expected above, a small number of points are distributed in cruise and most of them are distributed in “take-off” and “landing” segments, and the feature is especially striking in the hp and pk methods. This is because the locations of segments of an interval are based on the residual of solution in both hp and pk. It is seen intuitively that the same phenomenon in example 1 appears that $pk(3, 9, 3)$ method can achieve the solution meeting same tolerance requirements with fewer mesh nodes and intervals. The points and intervals in pk are much less than those in the other two methods, due to the locations of segments in pk method are optimized.

Next, to discuss the computational efficiency of the pk method for solving hyper-sensitive problem, table 2 shows the computational performance of solving hyper-sensitive problem for different values of t_f corresponding variable parameters. It is seen that the required number of collocation points are always minimal in solution solved by pk method. When $t_f = 40$ which is a small time range, the $ph(5, 15)$ is the least CPU time consuming method which only costs 10.0281 seconds. Because it is easy to make mesh dense including the parts of “take-off” and “landing”, when the total time range is small. And, the number of NLP parameters increased by adding nodes is not too much (only one

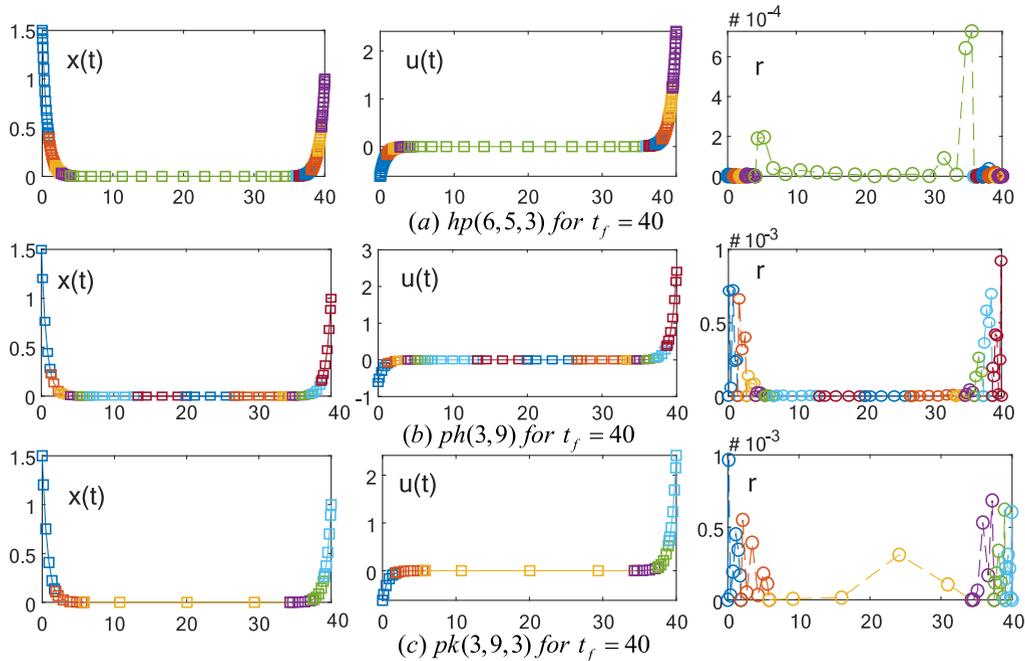


FIGURE 8. The state, control and residual vector on the final mesh grid of Hyper-Sensitive problem.

TABLE 2. Computational performance to solve Hyper-Sensitive problem.

Mesh refinement method	t_f	CPU time(s)	Collocation points	NLP parameters	Segments	No. of grids	e_{max}
<i>hp</i> (6, 5, 3)	40	86.2813	127	127*2	11	27	0.7271e-3
<i>hp</i> (10, 5, 3)	40	37.1563	119	119*2	9	16	0.6679e-3
<i>hp</i> (5, 10, 3)	40	28.7031	112	112*2	7	14	0.5502e-3
<i>ph</i> (3, 9)	40	15.0156	79	79*2	15	6	0.7176e-3
<i>ph</i> (4, 12)	40	11.9531	76	76*2	11	8	0.8142e-3
<i>ph</i> (5, 15)	40	10.0281	78	78*2	11	9	0.9143e-3
<i>pk</i> (3, 9, 3)	40	48.3750	55	55*2+9	10	8	0.9401e-3
<i>pk</i> (4, 12, 3)	40	36.3438	50	50*2+6	7	9	0.9441e-3
<i>pk</i> (5, 15, 3)	40	17.4688	60	60*2+5	6	8	0.6909e-3
<i>hp</i> (6, 5, 3)	1000	138.5781	121	242	13	32	0.5715e-3
<i>hp</i> (5, 10, 3)	1000	1.0424e+03	130	260	10	40	0.6201e-3
<i>ph</i> (3, 9)	1000	316.015	207	414	41	8	0.8128e-3
<i>ph</i> (5, 15)	1000	416.9531	214	418	28	14	0.9143e-3
<i>pk</i> (3, 9, 3)	1000	56.8906	66	143(=66*2+11)	12	10	0.9458e-3
<i>pk</i> (5, 15, 3)	1000	49.4688	69	144(=69*2+6)	7	15	0.7007e-3
<i>pk</i> (3, 9, 3)	5000	88.7656	79	173	16	12	0.7976e-3
<i>pk</i> (5, 15, 3)	5000	56.8906	75	157	8	16	0.9738e-3
<i>pk</i> (3, 9, 3)	10000	70.0313	83	182	17	15	0.8864e-3
<i>pk</i> (5, 15, 3)	10000	59.4844	84	177	10	20	0.7398e-3

state and one control variable for this problem, so one node produces two NLP parameters) which is not enough to affect the efficiency of NLP solver, but the locations of interval segments as parameters in the NLP solver cost more time. Also, it should be acceptable for the time cost in *pk*(5, 15, 3) with only 17.4688 seconds.

For more discussion, when t_f increases to 1000, the results are shown in the second part of table 2. In the *hp* methods, obvious defects were exposed that the time cost is sensitive for different parameters and much more iterations are required than other two methods to let solutions meet the predefined tolerance. In the *ph* methods, with the rise of t_f

from 40 to 1000, the number of collocation points increases greatly, resulting in a significant increase in the total time consumption of the *ph* methods. All above mentioned issues, such as parameter sensitivity, large number of iterations, and redundant points, are solved in the *pk* methods. For example, there are only 7 intervals and 69 collocation points in the last iterative mesh after 15 iterations with 49.4688 seconds for the solution of *pk*(5, 15, 3) when $t_f = 1000$. As an aside, it is interesting to note that the number of NLP parameters generated by *pk*(3, 9, 3) and *pk*(5, 15, 3) are almost the same (143 and 144 respectively), but *pk*(5, 15, 3) with more iterations has a less time cost. This is because there are only

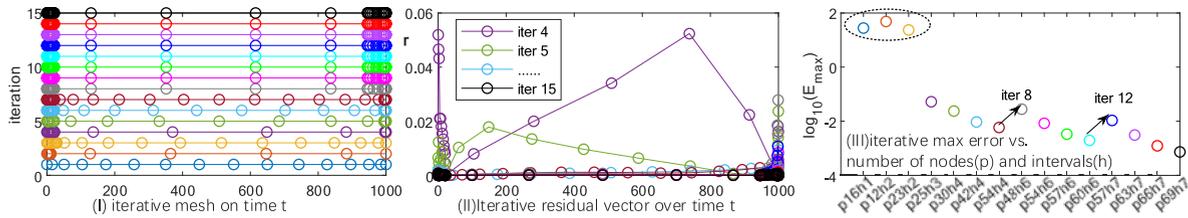


FIGURE 9. Convergence with $pk(5, 15, 3)$ mesh refinement of hyper-sensitive ($t_f = 1000$).

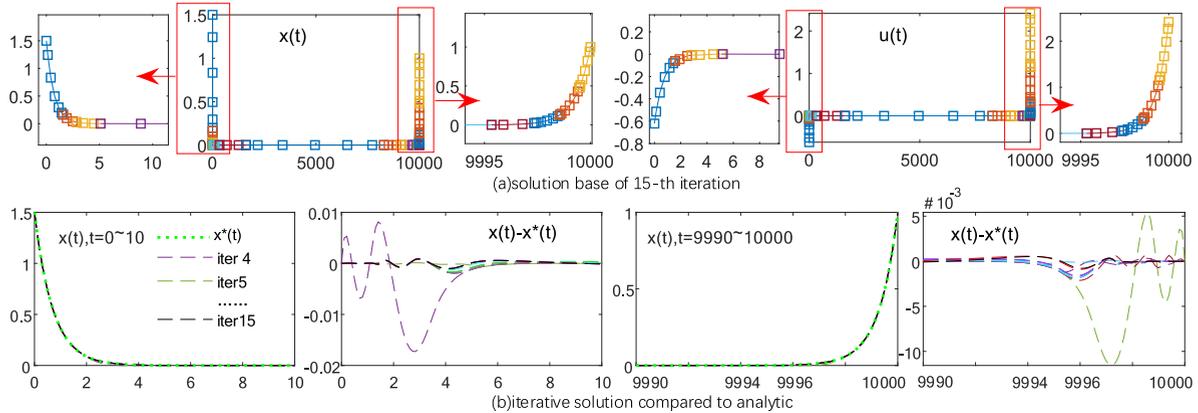


FIGURE 10. The solutions of hyper-sensitive using $pk(3, 9, 3)$ for $t_f = 10000$.

6 optimal knots in $pk(5, 15, 3)$, which is much less than 11 optimal knots in $pk(3, 9, 3)$. In other words, optimal knots as the parameters of NLP cost more time than state variables as the parameters of NLP under the same amount condition. This discovery can be used to better set the parameters of the pk method. Furthermore, we increased t_f to 5000 and 10000, and the pk method still performed well as shown in third part of table 2. The collocation points, number of intervals, and iterations remained almost unchanged in the case of t_f increasing by 5 and 10 times. In summary, the computational performance of pk methods is very stable for any $t_f \in [40, 1000, 5000, 10000]$, because the optimal knots can keep slow growth rates of collocation points and intervals when the t_f increases greatly.

Figure 9 shows the iterative residual vector, iterative max error, and the corresponded mesh distribution on each iteration of $pk(5, 15, 3)$ when $t_f = 1000$. Obviously, collocation points are gathered and constantly increased at the start and end of the mesh through iteration, corresponding to the “take-off” and “landing” segments of the problem. It is clear that the solution of the hyper-sensitives is constructed with three parts: “take-off”, “cruise” and “landing”. So, the numbers of intervals in the first three mesh grids are less than three resulting in solutions’ particularly large errors which are marked by a dotted ellipse in the part III plot and the corresponding residual vectors are difficult and to be plotted in part II of figure 9. From the fourth iteration, the residual converges to the small values, which are drawn as purple circle (or purple circle line) in the diagram figure 9. Also, there are twice accidents with error increasing at the 8-th and 12-th iteration. This is a normal phenomenon in our

method, because the collocation points of new intervals are set as the minimum number of collocation points, and errors may raise in a new interval, so as to reduce the total mesh size. As shown in (I), from 7-th mesh (garnet circle line) to 8-th mesh (gray circle line), the points number of “cruise” reduces to a smaller value, allowing the algorithm to focus more on the “take-off”, and “landing”. As shown in (III), there are 8 points reduced from “p54h4” in 7-th mesh to “p48h6” in 8-th mesh, and there are three points reduced from “p60h6” in 11-th mesh to “p57h7” in 12-th mesh. It can be seen that the number of nodes, number of intervals and location of interval segments are optimized in the pk method. This is why, after 15 iterations, only 69 points are required, much less than those in the other two methods (hp,ph) for $t_f = 1000$.

In addition, figure 10 shows the solutions of hyper-sensitive using $pk(3, 9, 3)$ after 15 iterations. It is easy to find the “take-off” and “landing” segments on the final mesh have the high density distribution points (colorful square) in figure 10(a) when the value of t_f is large enough, that is $t_f = 10000$. Only 83 collocation points are required, due to the collocation points’ optimal distribution. Furthermore, the analytic optimal state for this problem is given as

$$x^*(t) = \frac{e^{(t-t_f)\sqrt{2}}(1.5e^{-t_f\sqrt{2}} - 1) + (e^{-t_f\sqrt{2}} - 1.5)e^{-t\sqrt{2}}}{e^{-2t_f\sqrt{2}} - 1} \tag{34}$$

The more details about the accuracy of the solutions are given in figure 10 (b). Even if we zoom in to 10 seconds (0-10 and 9990-10000), it’s hard to see the difference

between numerical solutions and the analytic solution from the 4-th iteration. This is because the “take-off”, “cruise” and “landing” have been clearly divided into different intervals, the same as above discussion of solutions when $t_f = 1000$. Thereby, the difference between numerical solutions and the analytic solution is given by $x(t) - x^*(t)$, where $x(t)$ is calculated by interpolation equation (13), and then the difference of each iteration is clearly shown in the figure, which is decreased gradually with iteration.

VII. CONCLUSION

Mesh refinement bridges the gaps between theory and practice of the pseudospectral method for solving optimal control problem. A new mesh refinement strategy called pk-adaptive mesh refinement has been developed in the LGL pseudospectral method for improving the issues of redundant segments and nodes in state-of-art methods. The new iterative strategy for mesh refinement is devised that the degree of polynomial, the number of intervals and the locations of intervals within each mesh interval all are allowed to vary. The degree of polynomial is increased if the number of polynomial degree needed to estimate the mesh is less than the maximum allowable degree. Otherwise, the initial locations and number of knots are determined by the residual to refine the mesh. Then the locations of knots taken as the optimal parameters of optimal control problem are optimized simultaneously with other NLP variables.

The pk method has been applied to two examples to discuss the advantages compared with the hp and ph methods. In example 1, the moon-lander problem was solved in turn by three methods: hp, ph and pk. First, pk method obtains lower residual, fewer number of intervals and fewer number of nodes. Second, the discussions of the relation between convergence and mesh refining history have shown that the reason of why the pk method can quickly solve the lunar landing problem is that the optimal knots can accurately refine the mesh near the switch point of control variables in the bang-bang control problem. Third, the results of pk method under different parameters show the robustness of the method. In example 2, hyper-sensitive problem was solved by the three methods too. First, the results have shown that the solution of pk method can meet the same tolerance requirement with fewer mesh nodes and fewer intervals. Second, the analysis results of computational efficiency in solving hyper-sensitive problem show that pk method is more stable under different t_f values and method parameters. Especially, the efficiency of pk method is obviously better than the other two methods when t_f value is large. Third, the validity of the pk pseudo-spectral solution is further verified by comparing it with the analytical solution even if t_f is large enough.

In conclusion, the pk method has the advantages of higher computational efficiency and fewer number of collocation points with the same accuracy compared with the other two methods.

REFERENCES

- [1] N. Negarchi and K. Nouri, “A new direct method for solving optimal control problem of nonlinear Volterra–Fredholm integral equation via the Müntz–Legendre polynomials,” *Bull. Iranian Math. Soc.*, vol. 45, no. 3, pp. 917–934, 2019.
- [2] S. Tan, H. Lei, and T. Liu, “Optimal maneuver trajectory for hypersonic missiles in dive phase using inverted flight,” *IEEE Access*, vol. 7, pp. 63493–63503, 2019.
- [3] O. Abdelkhalik, S. Zou, R. Robinett, G. Bacelli, D. Wilson, and R. Coe, “Control of three degrees-of-freedom wave energy converters using pseudo-spectral methods,” *J. Dyn. Syst. Meas. Control-Trans.*, vol. 140, no. 7, 2018, Art. no. 074501.
- [4] S. T. Ul Islam Rizvi, H. Linshu, X. Dajun, and S. I. A. Shah, “Trajectory optimisation for a rocket-assisted hypersonic boost-glide vehicle,” *Aeronaut. J.*, vol. 121, no. 1238, pp. 469–487, 2017.
- [5] S. Mathavaraj, R. Pandiyan, and R. Padhi, “Constrained optimal multi-phase lunar landing trajectory with minimum fuel consumption,” *Adv. Space Res.*, vol. 60, no. 11, pp. 2477–2490, 2017.
- [6] L. N. Trefethen, *Spectral Methods in MATLAB*. Philadelphia, PA, USA: SIAM, 2000.
- [7] D. Garg, M. Patterson, W. W. Hager, A. V. Rao, D. A. Benson, and G. T. Huntington, “A unified framework for the numerical solution of optimal control problems using pseudospectral methods,” *Automatica*, vol. 46, no. 11, pp. 1843–1851, 2010.
- [8] G. Elnagar, M. A. Kazemi, and M. Razzaghi, “The pseudospectral Legendre method for discretizing optimal control problems,” *IEEE Trans. Autom. Control*, vol. 40, no. 10, pp. 1793–1796, Oct. 1995.
- [9] G. N. Elnagar and M. A. Kazemi, “Pseudospectral chebyshev optimal control of constrained nonlinear dynamical systems,” *Comput. Optim. Appl.*, vol. 11, no. 2, pp. 195–217, 1998.
- [10] K. Mamehrashi and A. Nematii, “A new approach for solving infinite horizon optimal control problems using Laguerre functions and Ritz spectral method,” *Int. J. Comput. Math.*, p. 16, Jun. 2019, doi: 10.1080/00207160.2019.1628949.
- [11] G. T. Huntington and A. V. Rao, “Optimal reconfiguration of spacecraft formations using the gauss pseudospectral method,” *J. Guid. Control Dyn.*, vol. 31, no. 3, pp. 689–698, 2012.
- [12] D. Garg, M. Patterson, C. Darby, C. Francolin, G. Huntington, W. Hager, and A. Rao, “Direct trajectory optimization and costate estimation of general optimal control problems using a radau pseudospectral method,” in *Proc. AIAA Guid., Navigat., Control Conf.* Chicago, IL, USA: AIAA, Aug. 2009, p. 5989.
- [13] R. Baltensperger and J.-P. Berrut, “The errors in calculating the pseudospectral differentiation matrices for Čebyšev–Gauss–Lobatto points,” *Comput. Math. Appl.*, vol. 37, no. 1, pp. 41–48, 1999.
- [14] F. Fahroo and I. M. Ross, “Costate estimation by a Legendre pseudospectral method,” *J. Guid., Control Dyn.*, vol. 24, no. 2, pp. 270–277, 2001.
- [15] M. Gyllenberg, F. Scarabel, and R. Vermiglio, “Equations with infinite delay: Numerical bifurcation analysis via pseudospectral discretization,” *Appl. Math. Comput.*, vol. 333, pp. 490–505, Sep. 2018.
- [16] F. Fahroo and I. M. Ross, “Advances in pseudospectral methods for optimal control,” in *Proc. AIAA Guid., Navigat. Control Conf. Exhib.* Honolulu, HI, USA: AIAA, Aug. 2008, p. 7309.
- [17] D. Garg, M. A. Patterson, C. Francolin, C. L. Darby, G. T. Huntington, W. W. Hager, and A. V. Rao, “Direct trajectory optimization and costate estimation of finite-horizon and infinite-horizon optimal control problems using a Radau pseudospectral method,” *Comput. Optim. Appl.*, vol. 49, no. 2, pp. 335–358, 2011.
- [18] M. A. Patterson and A. V. Rao, “GPOPS-II: A MATLAB software for solving multiple-phase optimal control problems using hp-adaptive Gaussian quadrature collocation methods and sparse nonlinear programming,” *ACM Trans. Math. Softw.*, vol. 41, no. 1, 2014, Art. no. 1.
- [19] I. M. Ross and F. Fahroo, “Legendre pseudospectral approximations of optimal control problems,” in *New Trends in Nonlinear Dynamics and Control and their Applications*, W. Kang, C. Borges, and M. Xiao, Eds. Berlin, Germany: Springer, 2003, pp. 327–342.
- [20] Q. Gong, F. Fahroo, and I. M. Ross, “Spectral algorithm for pseudospectral methods in optimal control,” *J. Guid., Control, Dyn.*, vol. 31, no. 3, pp. 460–471, May 2008.
- [21] C. L. Darby, W. Hager, and A. V. Rao, “An hp-adaptive pseudospectral method for solving optimal control problems,” *Optim. Control Appl. Methods*, vol. 32, no. 4, pp. 476–502, 2011.

[22] P. Liu, G. Li, X. Liu, L. Xiao, Y. Wang, C. Yang, and W. Gui, "A novel non-uniform control vector parameterization approach with time grid refinement for flight level tracking optimal control problems," *ISA Trans.*, vol. 73, pp. 66–78, Feb. 2018.

[23] S. Jain and P. Tsiotras, "Trajectory optimization using multiresolution techniques," *J. Guid., Control, Dyn.*, vol. 31, no. 5, pp. 1424–1436, 2008.

[24] I. Babuška and B. Q. Guo, "The h , p and h - p version of the finite element method; basis theory and applications," *Adv. Eng. Softw.*, vol. 15, nos. 3–4, pp. 159–174, 1992.

[25] Z. Yi and X. Ge, "Attitude maneuver of dual rigid bodies spacecraft using hp-adaptive pseudo-spectral method," *Int. J. Aeronaut. Space Sci.*, vol. 20, no. 1, pp. 214–227, 2019.

[26] R. Chai, A. Savvaris, and A. Tsourdos, "Fuzzy physical programming for space manoeuvre vehicles trajectory optimization based on hp-adaptive pseudospectral method," *Acta Astronautica*, vol. 123, pp. 62–70, Jun./Jul. 2016.

[27] X. Zeng, N. Cui, and J. Guo, "Planning of space robot based on hp-adaptive pseudospectral method," *Jiqiren/Robot*, vol. 40, no. 3, pp. 385–392, 2018, doi: 10.13973/j.cnki.robot.170461.

[28] N. Koeppen, I. M. Ross, L. C. Wilcox, and R. J. Proulx, "Fast mesh refinement in pseudospectral optimal control," *J. Guid. Control Dyn.*, vol. 42, no. 4, pp. 711–722, 2019.

[29] M. A. Patterson, W. W. Hager, and A. V. Rao, "A ph mesh refinement method for optimal control," *Optim. Control Appl. Methods*, vol. 36, no. 4, pp. 398–421, Aug. 2015.

[30] H. M. Lei, T. Liu, J. Li, and Z. P. Jiang, "Adaptive mesh refinement of hp pseudospectral method using size reduction," *Control Theory Appl.*, vol. 33, no. 8, pp. 1061–1067, 2016.

[31] I. M. Ross and F. Fahroo, "Pseudospectral knotting methods for solving nonsmooth optimal control problems," *J. Guid., Control, Dyn.*, vol. 27, no. 3, pp. 397–405, 2004.

[32] C. C. Françolin, D. A. Benson, W. W. Hager, and A. V. Rao, "Cstate approximation in optimal control using integral Gaussian quadrature orthogonal collocation methods," *Optim. Control Appl. Methods*, vol. 36, no. 4, pp. 381–397, 2015.



ZHIGUI LIU is currently the Executive Dean of the Graduate School, Southwest University of Science and Technology, and the Ph.D. Supervisor with the Institute of Electronic Engineering, China Academy of Engineering Physics. His research interests include the areas of control theory and control engineering.



ZHIQIN LIU received the M.S. degree in computer engineering from the University of Electronic Science and Technology, Chengdu, China, in 1994. She is currently a Professor with the School of Computer Science and Technology, Southwest University of Science and Technology, Mianyang, China. Her current research interests include scientific computing, computer-aided diagnosis, and high-performance computing.



QINGFENG WANG received the M.S. degree in computer science from the Southwest University of Science and Technology, Mianyang, China, in 2014, and the Ph.D. degree with the School of Software Engineering, University of Science and Technology of China, Hefei, China, in 2019. Her current research interests include scientific computing, computer-aided diagnosis, and machine learning.



JUN HUANG received the M.S. degree in computer science from the Southwest University of Science and Technology, Mianyang, China, in 2014. He is currently pursuing the Ph.D. degree with the Graduate School, China Academy of Engineering Physics, Mianyang. He is also a Teaching Assistant with the School of Computer Science and Technology, Southwest University of Science and Technology. His current research interests include optimal control, big data, scientific computing, and high-performance computing.



JIE FU received the M.S. degree in computer science from the University of Science and Technology of China, Hefei, China, in 2006. She is currently a Lecturer with the School of Computer Science and Technology, Southwest University of Science and Technology, Mianyang, China. Her current research interests include scientific computing, computer-aided diagnosis, and computer graphics.

...