

Research Article

An Integrated Multicriterion *hp*-Adaptive Pseudospectral Method for Direct Optimal Control Problems Solving

Hongfu Liu, Shaofei Chen, Lincheng Shen, and Jing Chen

College of Mechatronic Engineering and Automation, National University of Defense Technology, Hunan Province, Changsha 410073, China

Correspondence should be addressed to Hongfu Liu, liu.ho@163.com

Received 4 July 2012; Revised 5 October 2012; Accepted 6 October 2012

Academic Editor: Ming Li

Copyright © 2012 Hongfu Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Pseudospectral methods (PMs) for solving general optimal control problems (OCPs) attract an increasing amount of research and application in engineering. It is challenging to improve the convergence rate, the solution accuracy, and the applicability of PMs, especially for nonsmooth problems. Existing *hp*-adaptive PMs consider only one heuristic criterion, which cannot produce satisfactory performance in many cases. In this paper, we propose a novel method which integrates multicriterion to *hp*-adaptive PM, in order to further improve the performance. For this purpose, we first devise an OCP solving framework of *hp*-adaptive PM. We then design a multicriterion *hp*-adaptive strategy which introduces prior knowledge, intermediate error and curvature as useful criterions for adaptive refinement. We last present an iterative procedure for solving general nonlinear OCPs. Results from two examples show that our method significantly outperforms competitors on the convergence rate and the solution accuracy. The method is practical and effective for direct solving of various OCPs in a broad range of engineering.

1. Introduction

There exist a large number of optimal control problems (OCPs) in the engineering area, such as aircraft trajectory optimization, robotic control, space rendezvous and docking, lunar landing trajectory design, wave maker, energy production, fatigue test, and so on [1–9]. Numerical solving methods for OCPs fall into two categories which are indirect methods and direct methods, respectively. Indirect methods require rigidly necessary conditions which often cannot be obtained in engineering practice. In direct methods, an OCP is transcribed into a finite-dimensional nonlinear programming problem (NLP), in which the necessary conditions from the calculus of variations are not rigorous. Direct methods are becoming more and more popular for solving various OCPs with the well development of computer

technology [1]. As a major class of direct methods, pseudospectral method (PM) attracts more and more attention from academic research and engineering application [2]. For example, SIAM news reported that PM was successfully applied to optimal control and motion planning of the international space station in 2007. The application helped NASA to save about one million dollars in three hours [3].

The basic principle of PM is to approximate the state using a set of basis functions while a set of differential algebraic constraints are enforced at a finite number of collocation points. How to divide the element intervals and adjust the degree of basis function polynomial are two key components. Three kinds of approach are usually employed in order to achieve a specified error tolerance. The first one is named h -method, which uses many low-degree approximating subintervals. The terminology " h " denotes the method in which the degree of the elements is fixed and convergence is achieved by refining the mesh size h . The name of second one is p -method, which utilizes a few fixed numbers of intervals with a variable degree polynomial in each interval. The terminology " p " denotes the method fixes the mesh and achieves convergence by increasing the polynomial degree p of the elements. And the third approach is classified as hp -adaptive method, which allows for a variable number of approximating intervals with a variable degree approximation within each interval. The terminology " hp " denotes the method which allows for refinement in both the element size h , and the polynomial degree p simultaneously. The three approaches derive from the finite element method (FEM) in mechanics and fluid dynamics [10, 11], which are h -FEM, p -FEM, and hp -adaptive FEM, respectively.

Gauss PM (GPM) [12], Radau PM (RPM) [13], and Lobatto PM (LPM) [14] are three most well-developed PMs. The collocation points of them are based on accurate quadrature rules. Their basis functions are typically Chebyshev or Lagrange polynomial. They all belong to p -methods and they achieve convergence by increasing the degree of basis function polynomial. p -methods have a simple structure and converge at an exponential rate for problems which their solutions are infinitely smooth [15, 16]. However, many OCPs in engineering have either nonsmooth solutions or nonsmooth problem formulations. The nonsmooth feature results in low convergence in p -methods and leads to an extremely large NLP in h -methods.

It is natural to combine the strongpoint of h -methods and p -methods to form a class of hp -adaptive methods to improve the applicability of PM. One typical hp -adaptive method was proposed by Darby et al. in [17], their method is based on the relative curvature. If the ratio between the maximum curvature and the average curvature is sufficiently large, then the error is decreased by refining the mesh; the process is called h -refinement. Otherwise, the accuracy is improved by increasing the degree of approximating polynomial within an element; the process is named p -refinement. Another adaptive strategy decides h -refinement or p -refinement according to the convergence rate in each element [18]. The h -refinement is adopted only when exponential convergence is lost in an existing element. Based on the observation, single criterion is used by the aforementioned hp -adaptive methods. There are various widely used criteria, heuristic mechanisms, and adaptive strategies in the FEM area, which are of great help for us to design a more efficient hybrid adaptive algorithm.

hp -adaptive methods attract increasing amount of attentions in the research of FEM [11, 19–25]. Babuška discussed the mathematical properties of p - and hp -adaptive methods [10]. Galvão et al. addressed a hp -adaptive least-squares spectral element method (LSSEM) for solving hyperbolic partial differential equations [26]. Ben Dhia et al. proposed a new adaptive method based on an optimal control approach for adaptive modeling of an atomic-to-continuum coupling method constructed from the Arlequin framework [27]. In

[28], Oden and Prudhomme expanded the adaptive control of modeling error to include ideas of statistical calibration, validation, and uncertainty quantification. Dorao and Jakobsen gave a use case of applying *hp*-adaptive LS-SEM for the population balance equation [29]. An *hp*-adaptive spectral solver for reactor modeling was given in [30]. Lu investigated the adaptive mixed FEMs for parabolic optimal control problems and introduced an adaptive algorithm based on posteriori error estimates to guide the mesh refinement [31]. Mitchell and McClain summarized several *hp*-adaptive strategies in FEM [11]. These strategies include prior knowledge of solution regularity, regularity estimation, type parameter, Texas three steps strategy, prediction error, nonlinear programming, and so on. Texas three steps include initialization, adaptive *h*-refinement, and adaptive *p*-refinement [32]. The adaptive strategy uses intermediate error as a criterion, which reduces the number of iterations efficiently. An adaptive strategy based on the local regularity of the solution was proposed in [33], which includes a method for estimating the local regularity. The successful applications of various *hp*-adaptive strategies in the FEM research give us a good deal of enlightenment. The heuristic criterions and the adaptive strategies are valuable for developing an integrated multicriterion *hp*-adaptive PM for direct OCPs solving.

In this paper, we intend to integrate multicriterion to *hp*-adaptive PM, for the purpose of further enhancing the performance of computation and approximation. We first devise an OCP solving framework of *hp*-adaptive PM. We investigate the approximation method of PM and the assessment of approximation error. In the framework, we focus on the algorithm of adaptive strategy and refinement here, which is called *hp*-adaptive algorithm. We then design a multicriterion *hp*-adaptive strategy which introduces prior knowledge, intermediate error, and curvature as useful criterions. These heuristic criterions support both adaptive strategy and adaptive refinement. The criterions of intermediate error and curvature are complementary for adaptive strategy in various OCPs solving. After that strategy design, we present an iterative algorithmic procedure. Our method converges by increasing the polynomial degree in the smooth segments of a solution; meanwhile it adaptively refines the mesh for the nonsmooth segments in an efficient way. Our evaluation results show that the proposed method significantly outperforms *hp*-adaptive PM based on curvature and effectively improves the convergence rate of computation and the accuracy of solution.

The remainder of this paper is organized as follows. Section 2 defines multiple-interval nonlinear continuous OCP in Bolza form. Section 3 describes the OCP solving framework of *hp*-adaptive PM. Section 4 designs a novel multicriterion *hp*-adaptive strategy. Section 5 details our integrated multicriterion *hp*-adaptive PM. Section 6 provides two examples for illustrating our method. Finally, Section 7 concludes the paper.

2. Multiple-Interval Nonlinear Continuous Optimal Control Problem

Consider the following multiple-interval nonlinear continuous optimal control problem in Bolza form. Minimize the cost function

$$J = \phi(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} g(\mathbf{x}(t), \mathbf{u}(t), t) dt \quad (2.1)$$

subject to the dynamic constraints

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t), \quad (2.2)$$

the inequality path constraints

$$\mathbf{C}(\mathbf{x}(t), \mathbf{u}(t), t) \leq 0, \quad (2.3)$$

and the boundary conditions

$$\psi(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f) = 0, \quad (2.4)$$

where $\mathbf{x}(t) \in \mathbb{R}^m$ is the state, $\mathbf{u}(t) \in \mathbb{R}^n$ is the control, $\mathbf{C}(\mathbf{x}(t), \mathbf{u}(t), t) \in \mathbb{R}^p$ is the control and state constraints, t is time, t_0 is the start time, and t_f is the terminal time.

Next, consider dividing the nonlinear continuous optimal control problem defined above into K element intervals. Let $t_0 < t_1 < t_2 < \dots < t_K$, where $t_K = t_f$ is the terminal time. An elements k is defined to begin at the mesh point t_{k-1} and end at the mesh point t_k . The time domain in each element k , $t \in [t_{k-1}, t_k]$ is transformed to $\tau \in [-1, 1]$ by the formula

$$\tau = \frac{2t - (t_{k-1} + t_k)}{t_k - t_{k-1}}, \quad (t_{k-1} < t_k) \quad (2.5)$$

$\tau_0 \equiv -1$, and $\tau_f \equiv 1$. Furthermore, the inverse transformation is given as

$$t = \frac{[(t_k - t_{k-1})\tau + (t_{k-1} + t_k)]}{2}. \quad (2.6)$$

The nonlinear continuous-time optimal control problem on element $[t_0, t_f]$, which is defined by (2.1)–(2.4), can be expressed as a K element intervals problem. First, the cost function of (2.1) can be written as

$$J = \phi(\mathbf{x}^{(1)}(\tau_0), t_0, \mathbf{x}^{(K)}(\tau_f), t_K) + \sum_{k=1}^K \left[\frac{t_k - t_{k-1}}{2} \int_{\tau_0}^{\tau_f} g(\mathbf{x}^{(k)}(\tau), \mathbf{u}^{(k)}(\tau), \tau; t_0, t_K) d\tau \right]. \quad (2.7)$$

Next, the dynamics constraints, the boundary conditions, and the inequality path constraints are given, respectively, as

$$\frac{d\mathbf{x}^{(k)}(\tau)}{d\tau} = \frac{t_k - t_{k-1}}{2} \mathbf{f}(\mathbf{x}^{(k)}(\tau), \mathbf{u}^{(k)}(\tau), \tau; t_0, t_K), \quad (k = 1, 2, \dots, K) \quad (2.8)$$

$$\psi(\mathbf{x}^{(1)}(\tau_0), t_0, \mathbf{x}^{(K)}(\tau_f), t_K) = 0, \quad (2.9)$$

$$\frac{t_k - t_{k-1}}{2} \mathbf{C}(\mathbf{x}^{(k)}(\tau), \mathbf{u}^{(k)}(\tau), \tau; t_0, t_K) \leq 0, \quad (k = 1, 2, \dots, K) \quad (2.10)$$

and continuity constraints at element interfaces

$$\mathbf{x}^{(k)}(\tau_f) = \mathbf{x}^{(k+1)}(\tau_0), \quad (k = 1, 2, \dots, K - 1). \quad (2.11)$$

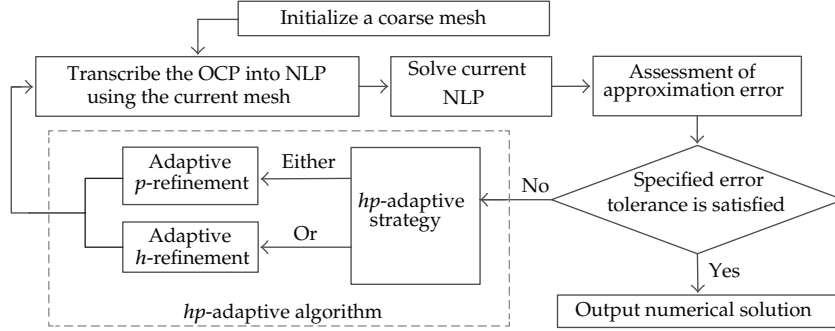


Figure 1: Flow chart of hp -adaptive pseudospectral method.

3. Framework of Direct Optimal Control Problems Solving Using hp -Adaptive Pseudospectral Method

3.1. Flow Chart of hp -Adaptive Pseudospectral Method

Pseudospectral method transcribes a continuous OCP into a nonlinear programming problem (NLP). The flow chart of hp -adaptive PM is described in Figure 1. The major steps include pseudospectral transcription, solving NLP, error estimate, and adaptive strategy and refinement. In this paper, we choose Sparse Nonlinear OPTimizer (SNOPT) as the NLP solver. Here we focus on the algorithm of adaptive strategy and refinement, which is called hp -adaptive algorithm.

hp -adaptive algorithm refines mesh and basis function adaptively, which is the key component of hp -adaptive PM. hp -adaptive algorithm involves hp -adaptive strategy, adaptive h -refinement, and adaptive p -refinement. During an iterative procedure, the algorithm assesses the approximation error of solution and analyses the curvature of state curve; and then hp -adaptive strategy decides which refinement (h -refinement or p -refinement) is more suitable. Adaptive h -refinement improves the accuracy of solution by refining the distribution of meshes and collocation points. And the goal of adaptive p -refinement is to achieve an exponential convergence rate by adjusting the degree of basis functions adaptively. In general, hp -adaptive algorithm integrates the strongpoint of both h -methods and p -methods to optimize the mesh and the polynomial degree of each element, which can simultaneously improve accuracy and computational efficiency.

3.2. Transcribe OCP Using Pseudospectral Method

In this paper, we choose the RPM [12, 34] as the foundation for transcribing a OCP into a NLP. For an element interval $[t_{k-1}, t_k], t_0 \equiv t_{k-1}, t_f \equiv t_k$, our method approximates the states and controls via the local interpolating polynomials with variable low degree. The state variables and the control variables in each element can be approximated by variable low-order Lagrange interpolating polynomials, which are expressed as

$$\mathbf{x}(\tau) \approx \mathbf{X}(\tau) = \sum_{i=0}^N \mathbf{X}(\tau_i) L_i(\tau), \quad (3.1)$$

$$\mathbf{u}(\tau) \approx \mathbf{U}(\tau) = \sum_{i=0}^N \mathbf{U}(\tau_i) L_i(\tau), \quad (3.2)$$

where (τ_1, \dots, τ_N) are collocation points, plus the start point $\tau_0 \equiv -1$, N is the number of collocation points. $L_i(\tau)$, $(i = 0, 1, \dots, N)$, is a basis of Lagrange polynomial with a variable degree. Then differential constraint of (2.8) can be transcribed into the algebraically form

$$\sum_{i=0}^N \dot{L}_i(\tau_i) x^i - \frac{t_f - t_0}{2} \mathbf{f}(\mathbf{X}_i, \mathbf{U}_i, \tau; t_0, t_f) = 0, \quad (3.3)$$

where $\mathbf{X}_i = \mathbf{X}(\tau_i)$, $\mathbf{U}_i = \mathbf{U}(\tau_i)$, $(i = 1, \dots, N)$. The initial state $\mathbf{X}_0 \equiv \mathbf{X}(\tau_0)$, and terminal state $\mathbf{X}_f \equiv \mathbf{X}(\tau_f)$ satisfies Gauss quadrature formula

$$\mathbf{X}_f \equiv \mathbf{X}_0 + \frac{t_f - t_0}{2} \sum_{i=1}^N \omega_i \mathbf{f}(\mathbf{X}_i, \mathbf{U}_i, \tau_i; t_0, t_f) = 0. \quad (3.4)$$

Differentiating $\mathbf{X}(\tau)$ in (3.1) with respect to τ , we get

$$\frac{d\mathbf{X}(\tau)}{d\tau} \equiv \dot{\mathbf{X}}(\tau) = \sum_{i=0}^N \mathbf{X}(\tau_i) \dot{L}_i(\tau) = \sum_{i=0}^N \mathbf{X}(\tau_i) \frac{dL_i(\tau)}{d\tau}. \quad (3.5)$$

The continuous objective function of (2.7) can be approximated via LGR quadrature

$$J \equiv \phi(\mathbf{X}_0, t_0, \mathbf{X}_f, t_f) + \frac{t_f - t_0}{2} \sum_{i=0}^N \omega_i g(\mathbf{X}_i, \mathbf{U}_i, \tau; t_0, t_f), \quad (3.6)$$

where ω_i is the weight coefficients of Gauss quadrature. And the boundary conditions of (2.9) is

$$\psi(\mathbf{X}_0, t_0, \mathbf{X}_f, t_f) = 0. \quad (3.7)$$

Combining the state and control variances, the trajectory is constrained by

$$\mathbf{C}(\mathbf{X}_i, \mathbf{U}_i, \tau_i; t_0, t_f) \leq 0. \quad (3.8)$$

Through the above process, a continuous Bolza OCP is transcribed into a limited dimensional NLP.

3.3. Assessment of Approximation Error

In the iterative procedure, assessment of approximation error is not only a stopping criterion, but also useful information for adaptive refinement. If local solution on a mesh element has not satisfy the specified accuracy, then the distribution of collocation points needs to be modified, either by increasing the polynomial degree on the element, or by redividing the element.

The state and control variables are approximated by basis function polynomials. For the k th element interval $[t_{k-1}, t_k]$ in mesh, ($k = 1, 2, \dots, K$), we firstly transform the interval of variable t into the interval of $\tau \in [\tau_0, \tau_f]$. Then the approximation polynomials of terminal state are given as

$$\begin{aligned}\widehat{\mathbf{X}}_0^k &= \mathbf{X}_0^{k-1} + \sum_{i=0}^N \mathbf{f}(\mathbf{X}_i, \mathbf{U}_i, \tau_i; t_{k-1}, t_k) \\ &= \mathbf{X}_0^{k-1} + \int_{\tau_0}^{\tau_f} \dot{\mathbf{X}}(\tau) d\tau - \int_{\tau_0}^{\tau_f} \dot{\mathbf{X}}(\tau) d\tau + \sum_{i=0}^N \mathbf{f}(\mathbf{X}_i, \mathbf{U}_i, \tau_i; t_{k-1}, t_k) \\ &= \mathbf{X}_0^k - \left[\int_{\tau_0}^{\tau_f} \left(\sum_{i=0}^N \mathbf{X}(\tau_i) \dot{L}_i(\tau) \right) d\tau - \sum_{i=0}^N \mathbf{f}(\mathbf{X}_i, \mathbf{U}_i, \tau_i; t_{k-1}, t_k) \right],\end{aligned}\quad (3.9)$$

where N is the number of collocation points in the k th element, ($i = 0, 1, \dots, N$). Hence the local error of the element is given as

$$\mathbf{X}_0^k - \widehat{\mathbf{X}}_0^k = \int_{\tau_0}^{\tau_f} \left(\sum_{i=0}^N \mathbf{X}(\tau_i) \dot{L}_i(\tau) \right) d\tau - \sum_{i=0}^N \mathbf{f}(\mathbf{X}_i, \mathbf{U}_i, \tau_i; t_{k-1}, t_k). \quad (3.10)$$

Furthermore, we can define the absolute error of the k th element interval as

$$\varepsilon_k = \frac{1}{h_k} \left| \int_{\tau_0}^{\tau_f} \left(\sum_{i=0}^N \mathbf{X}(\tau_i) \dot{L}_i(\tau) \right) d\tau - \sum_{i=0}^N \mathbf{f}(\mathbf{X}_i, \mathbf{U}_i, \tau_i; t_{k-1}, t_k) \right|, \quad (3.11)$$

where $h_k = t_k - t_{k-1}$ is the length of the k th element interval. And the maximum relative error over all state variable components j of the differential equations is given by

$$\eta^{(k)} = \max_j \frac{\varepsilon_k}{\omega_j}, \quad (3.12)$$

where

$$\omega_j = \frac{1}{N} \sum_{i=0}^N |x_{j,i}|. \quad (3.13)$$

ω_j is the even value of the j th state variable on all N collocation points. $\eta^{(k)}$ is the maximum relative error of all state variables on the k th element interval.

Let $\tilde{\eta}$ be a specified accuracy tolerance for the discretized differential algebraic constraints. If the maximum violation of the differential algebraic equations in the k th element $\eta^{(k)}$ is less than $\tilde{\eta}$, then the approximation of the k th element is considered to satisfy the accuracy tolerance.

4. Multicriterion *hp*-Adaptive Strategy Design

The local approximation error estimator indicates the element which should be refined, but it does not indicate whether the element is better to refine by an *h*-refinement or by a *p*-refinement. It is no longer sufficient to guide the adaptive refinement. A method for making that determination is called *hp*-adaptive strategy. We learn the lesson of adaptive strategy in FEM. Based on the observation, we propose three classes of useful criterion for *hp*-adaptive strategy: prior knowledge, intermediate error, and curvature.

First, the priori knowledge of solution regularity is useful for *hp*-adaptive strategy. As one criterion, it guides a *p*-refinement on the smooth segments and an *h*-refinement near singularities or flections of state curve. For example, the linear elliptic partial differential equations with smooth coefficients only have point singularities near corners of the boundary and where boundary conditions change.

The priori knowledge-based criterion requires an estimator, which evaluates the regularity value of intermediate approximate solution. Here we define the regularity value as r . A method for estimating the local regularity of a solution on a given mesh was proposed in [33], it indicated that

$$r \approx 1 + \alpha, \quad (4.1)$$

where α is the parameter of a priori estimate for the error. α can be obtained from the previous computation of the error estimator.

At the initial step of *hp*-adaptive strategy, the regularity value r_k of approximate solution on the k th element is readily available by initialization. Let p_k^0 and p_k , respectively, denote the initial degree and current degree of basis polynomial in the k element. If $p_k + 2 \leq r_k$, then *p*-refinement is performed, otherwise *h*-refinement is adopted. Furthermore, let $p_k^0 = r_k^0 - 2$ be an optimal choice for initial *p*-refinement.

Second, intermediate error can be adopted as a criterion for *hp*-adaptive strategy. Define an intermediate error

$$\eta_I = \gamma \tilde{\eta}, \quad (4.2)$$

where γ is a parameter generally ranging from 5 to 20, $\tilde{\eta}$ is a specified error tolerance, for example let $\tilde{\eta} = 0.01$ (1% error). An appropriate value of γ should make tradeoff between the criterion of intermediate error and the criterion of curvature. From a lot of experimentations, we find that an adopted γ between 5 and 20 effects good complement of the multicriterions.

For the element k in mesh, if the estimated error $\eta^{(k)} > \eta_I$, it needs detailed *h*-refinement. The criterion based on intermediate error should lead to distribute the error more even on every subinterval. It has been shown that the error is equidistributed over all the subintervals for an optimal mesh using *h*-refinement [11]. Hence, an optimal *h*-adaptive mesh distribution should simultaneously reduce the global error and distribute the local error of each element equably.

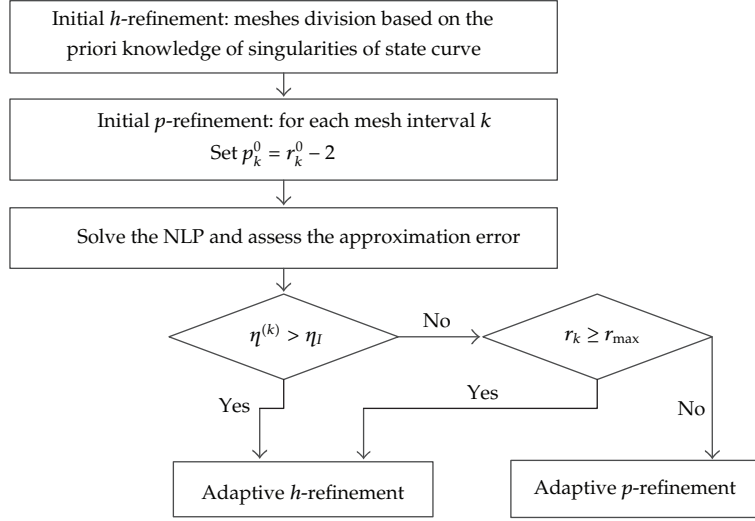


Figure 2: Flow chart of multicriterion hp -adaptive strategy.

Third, curvature is another criterion for hp -adaptive strategy. The curvature of state curve describes the regularity of intermediate solution. The curvature of the j th component of the state in element k is given by

$$\kappa^{(k)}(\tau) = \frac{|\dot{X}_j^{(k)}(\tau)|}{\left[1 + \dot{X}_j^{(k)}(\tau)^2\right]^{3/2}}, \quad (4.3)$$

where $X_j^{(k)}(\tau)$ is the j th component of the state approximation in the k th element [17]. Assume $\kappa_{\max}^{(k)}$ and $\bar{\kappa}^{(k)}$ be the maximum and mean value of $\kappa^{(k)}(\tau)$. Furthermore, the ratio of the maximum curvature to the mean curvature is defined

$$r_k = \frac{\kappa_{\max}^{(k)}}{\bar{\kappa}^{(k)}}. \quad (4.4)$$

For the element k in mesh, if $r_k < r_{\max}$ (where $r_{\max} > 0$ is a user-defined parameter), then p -refinement is adopted and a larger degree polynomial is used to obtain a better approximation. If $r_k \geq r_{\max}$, then h -refinement is adopted in this element.

Furthermore, we design an hp -adaptive strategy via combining prior knowledge, intermediate error and curvature as heuristic criteria. The flowchart of the integrated multicriterion hp -adaptive strategy is described in Figure 2. The priori knowledge of singularities or flections is adopted as heuristic criterion for initial h -refinement. The assessment of solution regularity is used to determine the initial degree of p -refinement. In the iterative procedure, the criteria of intermediate error and curvature are complementary for adaptive strategy in various OCPs solving.

5. Integrated Multicriterion hp -Adaptive Algorithm Procedure

In this section, we depict an integrated multicriterion hp -adaptive algorithm for a nonlinear continuous OCP with K intervals. The algorithm mainly includes six procedures.

Step 1 (initialization). Firstly, the error tolerance $\tilde{\eta}$ is specified, for example $\tilde{\eta} = 0.01$ (1% error). Then the intermediate error η_I is specified, for example, $\eta_I = 0.1$. And r_{\max} is specified, for example $r_{\max} = 8$.

Consider the initial h -refinement for each state equation, if the equation is a piecewise function, the initial h -refinement is located at the switch points of piecewise function; if there is a priori knowledge of singularities or flections of state curve, it guides the meshes division also. If there is no information about switch points or singularities, a coarse initial h -refinement is made with uniform mesh element size h_0 . The discrete approximation in each element is defined by the set of Legendre-Gauss-Radau points. And for the initial p -refinement, the polynomial-order of k th element is determined via the assessment of local solution regularity, it is given as $p_k^0 = r_k^0 - 2$.

Step 2 (solve the NLP using the current mesh). The problem is transcribed into the NLP on the current mesh, and then solving the NLP using sequential quadratic programming (SQP). A posteriori error estimator is used to estimate the approximation errors for the intermediate solution on each element, which is described in Section 3.3. The accuracy of the approximation in each element is assessed by calculating the differential-algebraic constraints on the collocation points.

Begin: For $k = 1, 2, \dots, K$.

Step 3. If $\eta^{(k)} \leq \tilde{\eta}$, then continue (proceed to next $k + 1$ element). $\eta^{(k)}$ is the approximation error of the k th element in mesh.

Step 4 (adaptive h -refinement). If either $\eta^{(k)} > \eta_I$, or $r_k \geq r_{\max}$, then h -refine the k th element. h -refinement keeps $p = p_0$ fixed and adaptively constructs elements with varying size. It maintains refinement until the solution achieves the intermediate error η_I and curvature ratio $r_k < r_{\max}$. The new number of element intervals is computed by formula

$$\begin{aligned}
 n_k &= \left\lceil \log_{10} \left(\frac{\eta^{(k)}}{\tilde{\eta}} \right) + c_h \times \log_e \left(\frac{\eta^{(k)}}{\eta_I} \right) \right\rceil \\
 &= \left\lceil \log_{10} \left(\frac{\eta^{(k)}}{\tilde{\eta}} \right) + c_h \times \log_e 10 \times \log_{10} \left(\frac{\eta^{(k)}}{\eta_I} \right) \right\rceil \\
 &= \left\lceil \log_{10} \left(\frac{\eta^{(k)}}{\tilde{\eta}} \right) + \log_{10} \left(\frac{\eta^{(k)}}{\eta_I} \right)^{c_h \times \log_e 10} \right\rceil \\
 &= \left\lceil \log_{10} \left(\left(\frac{\eta^{(k)}}{\tilde{\eta}} \right) \times \left(\frac{\eta^{(k)}}{\eta_I} \right)^{c_h \times \log_e 10} \right) \right\rceil,
 \end{aligned} \tag{5.1}$$

where $\eta^{(k)}/\tilde{\eta}$ is the ratio between current estimative maximum error and the specified error tolerance, $\eta^{(k)}/\eta_I$ is the ratio between current estimative maximum error and the intermediate error, and $\lceil \bullet \rceil$ is the operator that rounds to the next integer. c_h is an integer constant as a parameter for adjusting iterative speed, for example, $c_h = 2$. If there is some prior knowledge about state curve, c_h can be adjusted for flections.

Formula (5.1) determines a dissatisfactory element interval should be subdivided into how many subelements by h -refinement. The formula is based on two ratios. The first part depicts the approximation degree from the current error to the specified error tolerance; and the second part indicates the approximation degree from the current error to the intermediate error. They are calculated by the log operation, because h -refinement is hoped to attain an exponential convergence for the nonsmooth segment of function. The choices of logarithm are based on the quantity relation of the ratios. c_h controls the growth in the number of element intervals. If c_h is sufficiently large, the algorithm will use less iteration to converge to an acceptable solution, but the number of collocation points may increase quickly between iterations. If c_h is small, the mesh will increase slowly, but the algorithm may require many iterations. An appropriate value of c_h should make tradeoff between h -refinement and p -refinement.

The locations of the new required-elements are determined using the integral of a curvature density function in a manner similar to that given in [35]. Specifically, let $\rho(\tau)$ be the density function given by

$$\rho(\tau) = c\kappa(\tau)^{1/3}, \quad (5.2)$$

where c is a constant to satisfy

$$\int_{-1}^{+1} \rho(\zeta) d\zeta = 1. \quad (5.3)$$

The density function expressed by (5.2) is a probability density function (pdf) with fractional power-laws. Here we adopt 1/3 power in the function. In fact, from the point of view of power-laws in stochastic processes and/or fractal time series, 1/3 power is a typical heavy-tailed case. By heavy-tailed we mean that the density $\rho(\tau)$ decays slowly. And heavy-tailed pdfs imply that τ is in wild randomness due to infinite or very large variance [36].

Let $F(\tau)$ be the cumulative distribution function given by

$$F(\tau) = \int_{-1}^{\tau} \rho(\zeta) d\zeta. \quad (5.4)$$

The n_k new mesh points are chosen by

$$F(\tau) = \frac{i-1}{n_k}, \quad 1 \leq i \leq n_k + 1. \quad (5.5)$$

Finally, it is noted that if $n_k = 1$, then no subintervals are created. Therefore, the minimum value for n_k is 2.

Step 5 (adaptive p -refinement). p -refinement keeps mesh and the size of elements fixed. It adjusts the degree of polynomials on each element until the final error is satisfied. The degree of polynomial \hat{p}_k on the k th element is determined by

$$\begin{aligned}
\hat{p}_k &= p_k + \left[\log_{10} \left(\frac{\eta^{(k)}}{\tilde{\eta}} \right) + \log_e \left(\frac{\eta^{(k)}}{\eta_I} \right) \right] \\
&= p_k + \left[\log_{10} \left(\frac{\eta^{(k)}}{\tilde{\eta}} \right) + \log_e 10 \times \log_{10} \left(\frac{\eta^{(k)}}{\eta_I} \right) \right] \\
&= p_k + \left[\log_{10} \left(\frac{\eta^{(k)}}{\tilde{\eta}} \right) + \log_{10} \left(\frac{\eta^{(k)}}{\eta_I} \right)^{\log_e 10} \right] \\
&= p_k + \left[\log_{10} \left(\left(\frac{\eta^{(k)}}{\tilde{\eta}} \right) \times \left(\frac{\eta^{(k)}}{\eta_I} \right)^{\log_e 10} \right) \right],
\end{aligned} \tag{5.6}$$

where p_k is the initial degree of polynomial on this element. The growth in the polynomial degree is related to the log of the error ratio; because of a smooth function always exhibits exponential convergence in pseudospectral method.

End: For $k = 1, 2, \dots, K$.

Step 6 (return to Step 2). Our hp -adaptive algorithm integrates prior knowledge, intermediate error, and curvature for adaptive strategy and refinement, which is fit for smooth or nonsmooth problem. The illustration and comparison of our algorithm will be given by the examples in next section.

Remark 5.1. The accuracy and efficiency of the PMs for solving numerical optimal control problems have motivated the forefront of research. For continuous and smooth problems, pseudospectral knots and Gaussian quadrature rules are used to generate a natural spectral mesh that is dense near the points of interest. For nonsmooth problems, how to locate switches, kinks, corners, and other discontinuities is a challenge for solving practical OCPs. In our hp -adaptive algorithm procedure, adaptive h -refinement is devised to locate the knots where the solution is subject to sudden changes, as in points of discontinuity. In fact, from the point of view of stochastic processes, the curve of solution is higher order discontinuities in the control time history. According to its statistic properties, it is fractal time series. In general, fractal time series have a heavy-tailed probability distribution function, a slowly decayed autocorrelation function (ACF), and a power spectrum function (PSD) of $1/f$ type [37]. In our approach, we design the density function expressed by (5.2). We adopt the $1/3$ power, the density decays slowly, accordingly, heavy-tailed density. In our research of the practical OCP in engineering, such as aerospace engineering or mechanical engineering, the nonsmoothness and discontinuities are attributed to the differential dynamic and the constraint of system. The output or response of a differential system can be considered as a fractal time series [37, 38]. The density functions for mesh refinement in numerical optimal control are an instance of power-law-type pdf. The key problem of h -refinement is to find an appropriate density function. We mean that an appropriate choice of density function may help increase the accuracy of the solution and improve numerical robustness. Hence the theory of fractal

time series and power-laws would guide the choice. Li gives a tutorial review about fractal time series [37]. The locally weak stationarity of modified multifractional Gaussian noise is discussed in [39], which presents a set of stationary ranges. One-dimensional random function with long memory is introduced to model the sea level fluctuation [40]. Topics in power-laws are paid attention too, see for example, the contribution of approximating ideal filters by systems of fractional order in [41], the work of addressing power laws in cyber-physical networking systems in [36], the contribution of simplicial approach to fractal structures by Cattani et al. in [42], and the work of Li et al. in [38].

6. Numerical Examples

In this section, we present two numerical examples to illustrate the convergence rate and the applicability of the proposed method. The first example is a hypersensitive OCP, which solution has dramatic change on the “take-off” and “landing” segments. And the second example is a reusable launch vehicle re-entry trajectory optimization. The experimentations compare three algorithms: GPM [12], *hp*-adaptive PM based on curvature [17], and our integrated multicriterion *hp*-adaptive PM. For conciseness, the three algorithms are denoted by p , $hp^{(1)}$, $hp^{(2)}$, respectively. All CPU time recorded in table is whole time of iteration, which includes the time required to solve the NLP and the time required to refine the meshes. The program of these algorithms is expanded from open-source software GPOPS [43]. The transcribed NLP is solved by SNOPT [44, 45], using a 2.4-GHz Core 2 Duo, 2G DDR RAM personal computer with MATLAB R2009b. For all examples, the following values are chosen for the parameters described in Section 4: $r_{\max} = 8$ and $\gamma = 10$. In $hp^{(1)}$ and $hp^{(2)}$, the maximum number of collocation points on each mesh is 20, and the upper bound of iteration times is 20.

Example 6.1. Consider the following hypersensitive OCP adapted from [46]. It is a Hamiltonian boundary value OCP in long time interval. Minimize the cost function

$$J = \frac{1}{2} \int_0^{t_f} (x^2 + u^2) dt, \quad (6.1)$$

subject to the dynamic constraint

$$\dot{x} = -x^3 + u, \quad (6.2)$$

and the boundary conditions

$$x(0) = 1.5, \quad x(t_f) = 1, \quad (6.3)$$

where t_f is fixed. This problem has a “take-off”, “cruise”, and “landing” structure where all of the interesting behaviors occur in the “take-off” and “landing” segments. In this example, $t_f = 1000$.

The specification of initial grid is listed in Table 1. The efficiency and accuracy of various methods are listed in Table 2. Figure 3 shows the state and control curve of solution

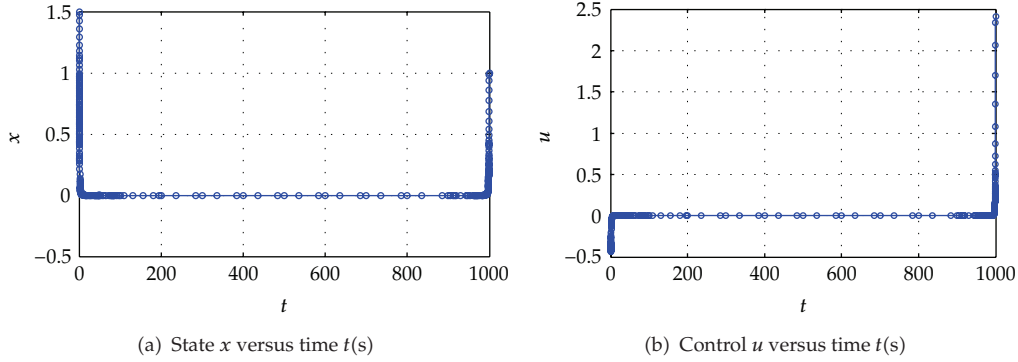


Figure 3: State and control curve on the final mesh of Example 6.1 for $hp^{(2)}$ -3 and $\tilde{\eta} = 1e - 3$.

Table 1: Initial grid used in Example 6.1.

Strategy	Number of collocation points	Number of intervals
p	60	1
$hp^{(1)}$ -5, $hp^{(2)}$ -5	10	5
$hp^{(1)}$ -3, $hp^{(2)}$ -3	10	3

on the final mesh for $hp^{(2)}$ -3 method, at the accuracy tolerance grade $\tilde{\eta} = 1e - 3$. The rings denote the location of collocation points. From analysis of these records, some results are illustrated.

First, hp -adaptive methods $hp^{(1)}$ and $hp^{(2)}$ outperform p -method in computational efficiency and approximation error. The results are better at all three accuracy tolerance grades. p -method even cannot achieve $1e - 3$ accuracy tolerance forever. Because p -method does not refine mesh, which results in the transcribed NLP is dense for this kind of nonsmooth problem. It incurs dramatic increase of computation time but low accuracy of approximation.

Second, the proposed $hp^{(2)}$ method is better than $hp^{(1)}$ method in computational efficiency. At all three accuracy tolerance grades, the count of collocation points for $hp^{(2)}$ -3, $hp^{(2)}$ -5 is less than the count of $hp^{(1)}$ -3, $hp^{(1)}$ -5, respectively. The use of prior knowledge can get an accurate initial mesh. The criterion of intermediate error optimizes the distribution of collocation points.

Third, the count of collocation points for hp -3 is less than the count of hp -5, but hp -3 method needs more iterations. From the comparison of experimental records, $hp^{(2)}$ -3 is the best method for solving this problem, which attains the fastest computational efficiency and the least approximation error.

Fourth, as shown in Figure 3, the state and control curves rapid change on the “take-off” and “landing” segments, where the relatively dense collocation points are distributed via the hp -adaptive algorithm. Meanwhile, the distribution of collocation points on the smooth segments is sparse. Compared with p -method and $hp^{(1)}$ method, our $hp^{(2)}$ method improves computational efficiency and reduces approximation error for this kind of nonsmooth OCP solving.

Example 6.2. Reusable launch vehicle re-entry trajectory optimization is an OCP of maximizing the cross range during the atmospheric entry [1]. Minimize the cost function

$$J = -\phi(t_f), \quad (6.4)$$

Table 2: Summary of results for Example 6.1 using various collocation strategies and accuracy tolerances.

$\tilde{\eta}$	Strategy	Approximate error	Collocation points	Elements	Mesh iterations	CPU time (s)
$1e-2$	p	$9.23e-2$	203	1	50	197.36
	$hp^{(1)-5}$	$6.51e-2$	366	72	8	66.90
	$hp^{(2)-5}$	$2.92e-2$	269	52	8	25.49
	$hp^{(1)-3}$	$5.03e-2$	337	102	10	49.67
	$hp^{(2)-3}$	$1.89e-2$	244	68	9	21.08
$1e-3$	p	—	—	—	—	—
	$hp^{(1)-5}$	$8.63e-3$	478	93	8	91.85
	$hp^{(2)-5}$	$4.10e-3$	347	66	7	42.93
	$hp^{(1)-3}$	$6.45e-3$	436	125	11	82.72
	$hp^{(2)-3}$	$2.17e-3$	329	87	9	39.80
$1e-4$	p	—	—	—	—	—
	$hp^{(1)-5}$	$7.14e-4$	614	119	9	117.22
	$hp^{(2)-5}$	$6.59e-4$	426	75	8	66.08
	$hp^{(1)-3}$	$6.97e-4$	565	152	11	107.79
	$hp^{(2)-3}$	$3.86e-4$	407	108	10	56.16

subject to the differential dynamic constraint

$$\begin{aligned}
 \dot{r} &= v \sin \gamma, & \dot{v} &= \frac{D}{m} - g \sin \gamma, \\
 \dot{\theta} &= \frac{v \cos \gamma \sin \psi}{r}, & \dot{\gamma} &= L \frac{\cos \sigma}{mv} - \left(\frac{g}{v} - \frac{v}{r} \right) \cos \gamma, \\
 \dot{\phi} &= \frac{v \cos \gamma \cos \psi}{r}, & \dot{\psi} &= \frac{L \sin \sigma}{mv \cos \gamma} + \frac{v \cos \gamma \sin \psi \tan \phi}{r},
 \end{aligned} \tag{6.5}$$

and the boundary conditions

$$\begin{aligned}
 r(0) &= 79248 + R_e \text{ m}, & r(t_f) &= 24384 + R_e \text{ m}, \\
 \theta(0) &= 0 \text{ deg}, & \theta(t_f) &= \text{Free}, \\
 \phi(0) &= 0 \text{ deg}, & \phi(t_f) &= \text{Free}, \\
 v(0) &= 7803 \text{ m/s}, & v(t_f) &= 762 \text{ m/s}, \\
 \gamma(0) &= -1 \text{ deg}, & \gamma(t_f) &= -5 \text{ deg}, \\
 \psi(0) &= 90 \text{ deg}, & \psi(t_f) &= \text{Free},
 \end{aligned} \tag{6.6}$$

where r is the geocentric radius, θ is the longitude, ϕ is the latitude, v is the speed, γ is the flight path angle, ψ is the azimuth angle, and σ is the bank angle. The radius of the earth $R_e = 6371203.92 \text{ m}$.

The specification of initial grid is listed in Table 3. The efficiency and accuracy of various methods are listed in Table 4. Figure 4 shows the state and control curve of solution

Table 3: Initial grid used in Example 6.2.

Strategy	Number of collocation points	Number of intervals
p	50	1
$hp^{(1)}-6, hp^{(2)}-6$	10	6
$hp^{(1)}-4, hp^{(2)}-4$	10	4

Table 4: Summary of results for Example 6.2 using various collocation strategies and accuracy tolerances.

$\tilde{\eta}$	Strategy	Approximate error	Collocation points	Elements	Mesh iterations	CPU time (s)
$1e-3$	p	$7.61e-3$	58	1	4	70.88
	$hp^{(1)}-6$	$5.12e-3$	90	18	2	19.55
	$hp^{(2)}-6$	$3.55e-3$	70	12	3	18.05
	$hp^{(1)}-4$	$6.81e-3$	113	32	4	32.55
	$hp^{(2)}-4$	$4.19e-3$	83	18	4	19.66
$1e-4$	p	$9.17e-4$	75	1	12	88.94
	$hp^{(1)}-6$	$4.08e-4$	151	29	6	33.76
	$hp^{(2)}-6$	$3.23e-4$	122	21	4	24.89
	$hp^{(1)}-4$	$7.99e-4$	202	56	6	63.71
	$hp^{(2)}-4$	$5.83e-4$	153	34	4	40.24
$1e-5$	p	$9.74e-5$	90	1	16	99.50
	$hp^{(1)}-6$	$8.02e-5$	214	40	6	63.64
	$hp^{(2)}-6$	$7.50e-5$	183	32	5	53.44
	$hp^{(1)}-4$	$8.93e-5$	293	70	5	89.46
	$hp^{(2)}-4$	$7.69e-5$	251	51	5	84.88

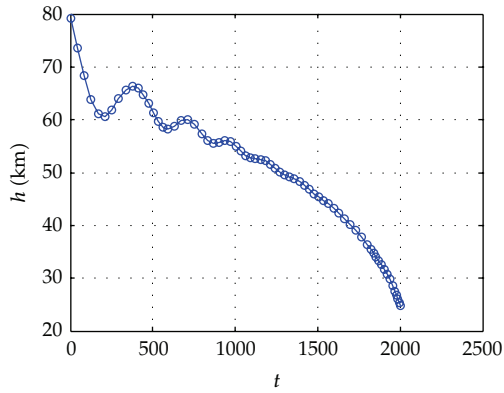
on the final mesh for $hp^{(2)}-6$ method, at the accuracy tolerance grade $\tilde{\eta} = 1e-4$. The state and control curves are relatively smooth for this example. From analysis of these records, some results are illustrated.

First, hp -adaptive methods $hp^{(1)}$ and $hp^{(2)}$ are similar or better than p -method in this example. p -method can achieve $1e-5$ accuracy tolerance and has the fewest number of collocation points, but it needs much iteration. The computational time of p -method is longer than hp -adaptive methods. The low computational efficiency of p -method is due to the fact that the transcribed NLP is dense.

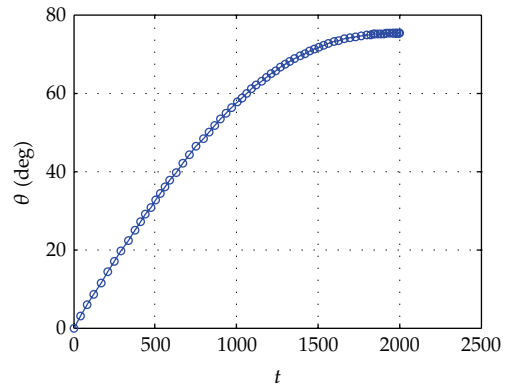
Second, our $hp^{(2)}$ method outperforms $hp^{(1)}$ method in computational efficiency. The count of collocation points for $hp^{(2)}$ is less than the count of $hp^{(1)}$ at three accuracy tolerance grades. The comparison illustrates that the distribution of collocation points via $hp^{(2)}$ is better than the distribution via $hp^{(1)}$.

Third, the count of collocation points for $hp-6$ is less than the count of $hp-4$. Based on the comparison of experimental records, $hp^{(2)}-6$ is the best method for solving this problem, which has the fastest computational efficiency and the least approximation error. In general, our method is feasible and improves computational efficiency for this kind of smooth OCP solving.

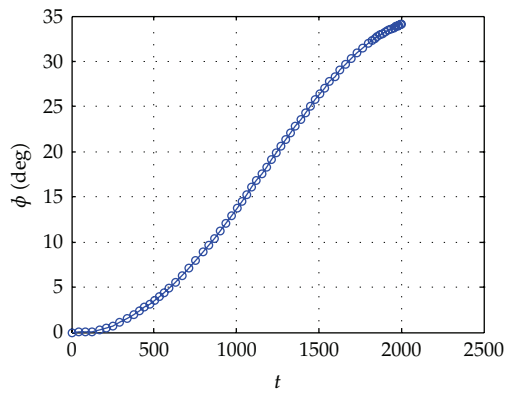
Remark 6.3. The results of examples highlight several points of p -methods and hp -adaptive methods. p -methods are fit for solving problems which their solutions are smooth. hp -adaptive methods perform similarly or better to p -methods for smooth problems. For nonsmooth problems, p -methods cannot accurately capture the discontinuities and rapid



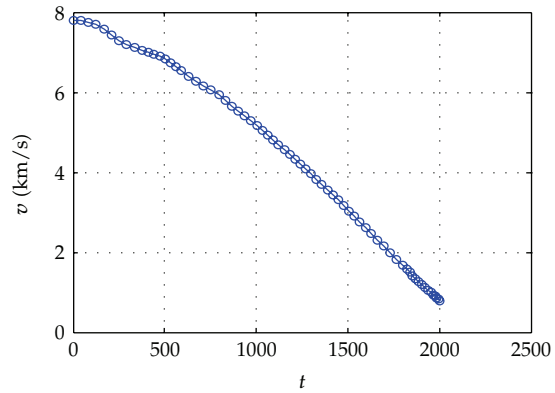
(a) Height h (km) versus time t (s)



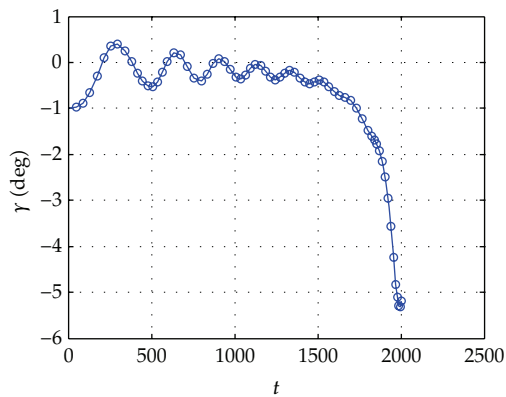
(b) Longitude θ (deg) versus time t (s)



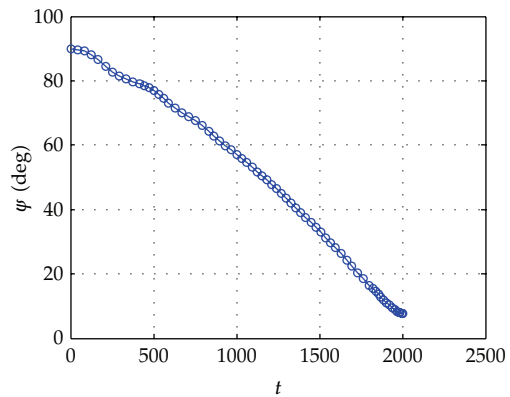
(c) Latitude ϕ (deg) versus time t (s)



(d) Velocity v (km/s) versus time t (s)



(e) Flight path angle γ (deg) versus time t (s)



(f) Azimuth angle ψ (deg) versus time t (s)

Figure 4: State and control curve on the final mesh of Example 6.2 for $hp^{(2)}$ -6 and $\tilde{\eta} = 1e - 4$.

changes in the trajectory. They only achieve low-accuracy even with much iteration. When a high-accuracy tolerance is required to satisfy, *hp*-adaptive methods are the likeliest candidates. They transcribe the OCPs into the NLPs with computational sparsity, which leads to the reduction of execution time. Compared with *hp*-adaptive PM based on curvature, our method is more efficient in the convergence rate; meanwhile the solutions achieve a comparable or better accuracy. The prior knowledge and the criterion of intermediate error leverage the enhancement of performance. The prior knowledge guides an accurate initial mesh, and the criterion of intermediate error optimizes the distribution of collocation points.

7. Conclusions

PMs become increasingly popular for direct OCPs solving in the engineering area. It is challenging to improve the convergence rate, the solution accuracy, and the applicability of algorithms, especially for nonsmooth problems. In this paper, we proposed a novel method which integrates multicriterion to *hp*-adaptive PM, in order to further improve the performance of computation and approximation. For this purpose, we first devised an OCP solving framework of *hp*-adaptive PM. We then designed a multicriterion *hp*-adaptive strategy which introduces prior knowledge, intermediate error, and curvature as useful criterions for adaptive refinement. The criterions of intermediate error and curvature were complementary for adaptive strategy in various OCPs solving. We last proposed an iterative procedure for solving general nonlinear OCPs. Our method converged by increasing the polynomial degree in the smooth segments of a solution; meanwhile it adaptively refined the mesh for the nonsmooth segments in an efficient way. Results from examples showed that our method significantly outperforms *hp*-adaptive PM based on curvature and effectively improves the convergence rate of computation and the accuracy of solution. Meanwhile, it enhances the applicability of *hp*-adaptive PMs for solving various OCPs.

For the future work, we will evaluate the performance of the proposed method using more OCPs in the engineering area. We will also try to exploit other valuable criterions for *hp*-adaptive method, which may further improve the convergence rate and the applicability of PMs. In addition, the choices of initial grid of PMs and the values of parameters (γ , r_{\max} , and c_h) will be further researched for various problems. Different problem types will need dissimilar parameters for fast solving. Another topic is applying fractal time series and power-laws to guide the choice of density function for various problems. This point itself is an issue worth discussing in our future work.

Appendix

Basis of the Pseudospectral Method

Much of the details of pseudospectral methods (PMs) are extensively described elsewhere [2, 15, 16, 34]; here, we briefly summarize the basis of the PM. The basic principle of a PM is to solve the continuous optimal control problem (OCP) by discretizing it to the NLP problem. The whole process mainly includes three steps. First, the horizon of a continuous OCP is mapped to a finite spectral interval $[-1, 1]$. Second, a PM is applied to discretize the problem. The discretized problem generates a sequence of approximate problems parameterized by the size of the grid. In the third step, each discretized problem in this sequence is solved by a globally convergent sequential quadratic programming (SQP) technique. In the PMs,

the functions are approximated by Lagrange interpolating polynomials of order N in which interpolation occurs at Gaussian quadrature points. There are three different families of Gauss quadrature points known as Legendre-Gauss, Legendre-Gauss-Radau, and Legendre-Gauss-Lobatto. Gauss PM (GPM) [12], Radau PM (RPM) [13], and Lobatto PM (LPM) [14] are the three most well-developed PMs. The collocation points of them are based on accurate quadrature rules. Their basis functions are typically Chebyshev or Lagrange polynomial.

Here we describe the procedure of GPM as an illustrative method. The GPM approximates the state using a basis of global interpolating Lagrange polynomials. These global polynomials are based on a set of discrete Legendre-Gauss (LG) points across the interval.

The standard interval considered here is denoted as $\tau \in [-1, 1]$. By using a linear transformation, the actual time t can be expressed as a function of τ via

$$t = \frac{[(t_f - t_0)\tau + (t_f + t_0)]}{2}, \quad \tau = \frac{2t}{(t_f - t_0)} - \frac{(t_f + t_0)}{(t_f - t_0)}, \quad (\text{A.1})$$

where t_0 and t_f stand for the initial and final time, respectively.

The state is approximated using a basis of $N + 1$ Lagrange interpolating polynomials, $L_i(\tau)$ ($i = 0, \dots, N$), as follows:

$$\mathbf{x}(\tau) \approx \mathbf{X}(\tau) = \sum_{i=0}^N \mathbf{X}(\tau_i) L_i(\tau), \quad (\text{A.2})$$

where

$$L_i(\tau) = \prod_{j=0, j \neq i}^N \frac{\tau - \tau_j}{\tau_i - \tau_j}, \quad (\text{A.3})$$

where τ_i , ($i = 0, \dots, N$) are the LG points belonging to the set.

An approximation to the control, $\mathbf{U}(\tau)$, is formed with a basis of N Lagrange interpolating polynomials, $\tilde{L}_i(\tau)$ ($i = 1, \dots, N$), as follows:

$$\mathbf{u}(\tau) \approx \mathbf{U}(\tau) = \sum_{i=1}^N \mathbf{U}(\tau_i) \tilde{L}_i(\tau). \quad (\text{A.4})$$

The differential dynamic constraints should be posed to ensure that the entire discrete state satisfies the dynamic equations. The left-hand side of the dynamic equations is approximated by differentiating the state approximation of (A.1) at the LG points as follows:

$$\dot{\mathbf{x}}(\tau_k) \approx \dot{\mathbf{X}}(\tau_k) = \sum_{i=0}^N \mathbf{X}(\tau_i) \dot{L}_i(\tau_k) = \sum_{i=0}^N D_{ki} \mathbf{X}(\tau_i), \quad (k = 1, \dots, N). \quad (\text{A.5})$$

The differentiation matrix, $D \in \mathbb{R}^{N \times (N+1)}$, can be determined offline from dynamic equations:

$$D_{ki} = \dot{L}_i(\tau_k) = \begin{cases} \frac{(1 + \tau_k)\dot{P}_N(\tau_k) + P_N(\tau_k)}{(\tau_k - \tau_i)[(1 + \tau_i)\dot{P}_N(\tau_i) + P_N(\tau_i)]}, & i \neq k \\ \frac{(1 + \tau_i)\dot{P}_N(\tau_i) + 2\dot{P}_N(\tau_i)}{2[(1 + \tau_i)\dot{P}_N(\tau_i) + P_N(\tau_i)]}, & i = k. \end{cases} \quad (\text{A.6})$$

The N collocation equations require (A.5) to be equal to the right-hand side of the dynamic equations at the collocation points:

$$\sum_{i=0}^N D_{ki} \mathbf{X}(\tau_i) - \frac{t_f - t_0}{2} f(\mathbf{X}(\tau_k), \mathbf{U}(\tau_k), \tau_k; t_0, t_f) = 0, \quad (k = 1, \dots, N). \quad (\text{A.7})$$

As it is shown shortly, collocating strictly on the interior of the interval leads to a unique mathematical equivalence used to approximate the costate.

Combining the state and control variances, the constraints are discretized as follows:

$$\mathbf{C}(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f) \leq 0. \quad (\text{A.8})$$

Lastly, the integral term in the cost functional can be approximated with a Gauss quadrature as follows:

$$J = \Phi(\mathbf{X}_0, t_0, \mathbf{X}_f, t_f) + \frac{t_f - t_0}{2} \sum_{k=1}^N \omega_k g(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f), \quad (\text{A.9})$$

where ω_k is the weight coefficient of Gauss quadrature.

Through the above process, a continuous OCP is transcribed into a limited dimensional NLP. The formulation is as follows:

$$\begin{aligned} \min \quad & F(\mathbf{y}) \\ \text{s.t.} \quad & g_j(\mathbf{y}) \leq 0, \quad j = 1, 2, \dots, p \\ & h_k(\mathbf{y}) = 0, \quad k = 1, 2, \dots, q. \end{aligned} \quad (\text{A.10})$$

The transcribed NLP can be solved by well-developed algorithms, such as SQP, interior point method, particle swarm optimization and so on.

Acknowledgments

Great thanks go to the reviewers for valuable comments on our revision of the paper. This work was supported by the National Natural Science Foundation of China (61005077), the National Basic Research Program of China (6138101001), and the National Defense Foundation of China (403060103).

References

- [1] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, Advances in Design and Control, SIAM Press, Philadelphia, Pa, USA, 2nd edition, 2009.
- [2] 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.
- [3] K. W. and B. N., "Pseudospectral optimal control theory makes debut flight-saves NASA \$1 M in under 3 hrs," *SIAM News*, vol. 40, no. 7, pp. 1–2, 2007.
- [4] M. Li, "An optimal controller of an irregular wave maker," *Applied Mathematical Modelling*, vol. 29, no. 1, pp. 55–63, 2005.
- [5] S. Castellucci, M. Carlini, M. Guerrieri, and T. Honorati, "Stability and control for energy production parametric dependence," *Mathematical Problems in Engineering*, vol. 2010, Article ID 842380, 21 pages, 2010.
- [6] M. Soler, A. Olivares, and E. Staffetti, "Hybrid optimal control approach to commercial aircraft trajectory planning," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 3, pp. 985–991, 2010.
- [7] L. Xie, P. Yang, T. Yang et al., "Dual-EKF-based real-time celestial navigation for lunar rover," *Mathematical Problems in Engineering*, vol. 2012, Article ID 578719, 16 pages, 2012.
- [8] S. H. Pourtakdoust, M. Kiani, and A. Hassanpour, "Optimal trajectory planning for flight through microburst wind shears," *Aerospace Science and Technology*, vol. 15, no. 7, pp. 567–576, 2011.
- [9] M. Li, "An iteration method to adjusting random loading for a laboratory fatigue test," *International Journal of Fatigue*, vol. 27, no. 7, pp. 783–789, 2005.
- [10] I. Babuška and M. Suri, "The p and h - p versions of the finite element method, basic principles and properties," *SIAM Review*, vol. 36, no. 4, pp. 578–632, 1994.
- [11] W. F. Mitchell and M. A. McClain, "A survey of hp -adaptive strategies for elliptic partial differential equations," in *Recent Advances in Computational and Applied Mathematics*, pp. 227–258, Springer, Amsterdam, The Netherlands, 2011.
- [12] D. A. Benson, G. T. Huntington, T. P. Thorvaldsen, and A. V. Rao, "Direct trajectory optimization and costate estimation via an orthogonal collocation method," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 6, pp. 1435–1440, 2006.
- [13] D. Garg, M. A. Patterson, C. L. Darby et al., "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, 2011.
- [14] G. Elnagar, M. A. Kazemi, and M. Razzaghi, "The pseudospectral Legendre method for discretizing optimal control problems," *IEEE Transactions on Automatic Control*, vol. 40, no. 10, pp. 1793–1796, 1995.
- [15] B. Fornberg, *A Practical Guide to Pseudospectral Methods*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, New York, NY, USA, 1998.
- [16] Q. Gong, F. Fahroo, and I. M. Ross, "Spectral algorithm for pseudospectral methods in optimal control," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 3, pp. 460–471, 2008.
- [17] C. L. Darby, W. W. Hager, and A. V. Rao, "Direct trajectory optimization using a variable low-order adaptive pseudospectral method," *Journal of Spacecraft and Rockets*, vol. 48, no. 3, pp. 433–445, 2011.
- [18] 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, 2011.
- [19] A. K. Patra, A. Laszloffy, and J. Long, "Data structures and load balancing for parallel adaptive hp finite-element methods," *Computers and Mathematics with Applications*, vol. 46, no. 1, pp. 105–123, 2003.
- [20] L. Demkowicz, W. Rachowicz, and Ph. Devloo, "A fully automatic hp -adaptivity," *Journal of Scientific Computing*, vol. 17, no. 1–4, pp. 117–142, 2002.
- [21] L. Demkowicz, "Computing with hp -adaptive finite elements," in *One and Two Dimensional Elliptic and Maxwell Problems*, Chapman & Hall/CRC, Boca Raton, Fla, USA, 2007.
- [22] P. Solín, J. Červený, and I. Doležel, "Arbitrary-level hanging nodes and automatic adaptivity in the hp -FEM," *Mathematics and Computers in Simulation*, vol. 77, no. 1, pp. 117–132, 2008.
- [23] R. Tews and W. Rachowicz, "Application of an automatic hp adaptive finite element method for thin-walled structures," *Computer Methods in Applied Mechanics and Engineering*, vol. 198, no. 21–26, pp. 1967–1984, 2009.
- [24] P. Solín and M. Kuraz, "Solving the nonstationary Richards equation with adaptive hp -FEM," *Advances in Water Resources*, vol. 34, no. 9, pp. 1062–1081, 2011.
- [25] A. Szymczak, A. Paszýnska, M. Paszýnski et al., "Anisotropic 2D mesh adaptation in hp -adaptive FEM," *Procedia Computer Science*, vol. 4, pp. 1818–1827, 2011.

- [26] L. Galvão, M. Gerritsma, and B. D. Maerschalck, “*hp*-adaptive least squares spectral element method for hyperbolic partial differential equations,” *Journal of Computational and Applied Mathematics*, vol. 215, no. 2, pp. 409–418, 2008.
- [27] H. Ben Dhia, L. Chamoin, J. T. Oden, and S. Prudhomme, “A new adaptive modeling strategy based on optimal control for atomic-to-continuum coupling simulations,” *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 37–40, pp. 2675–2696, 2011.
- [28] J. T. Oden and S. Prudhomme, “Control of modeling error in calibration and validation processes for predictive stochastic models,” *International Journal for Numerical Methods in Engineering*, vol. 87, no. 1–5, pp. 262–272, 2011.
- [29] C. A. Dorao and H. A. Jakobsen, “*hp*-adaptive least squares spectral element method for population balance equations,” *Applied Numerical Mathematics*, vol. 58, no. 5, pp. 563–576, 2008.
- [30] C. A. Dorao, M. Fernandez, H. A. Jakobsen, and H. F. Svendsen, “*hp*-Adaptive spectral element solver for reactor modeling,” *Chemical Engineering Science*, vol. 64, no. 5, pp. 904–911, 2009.
- [31] Z. Lu, “Adaptive mixed finite element methods for parabolic optimal control problems,” *Mathematical Problems in Engineering*, vol. 2011, Article ID 217493, 21 pages, 2011.
- [32] J. T. Oden and A. Patra, “A parallel adaptive strategy for *hp* finite element computations,” *Computer Methods in Applied Mechanics and Engineering*, vol. 121, no. 1–4, pp. 449–470, 1995.
- [33] M. Ainsworth and B. Senior, “An adaptive refinement strategy for *hp*-finite element computations,” *Applied Numerical Mathematics*, vol. 26, no. 1–2, pp. 165–178, 1998.
- [34] G. T. Huntington, *Advancement and Analysis of a Gauss Pseudospectral Transcription for Optimal Control Problems*, Massachusetts Institute of Technology, Cambridge, Mass, USA, 2007.
- [35] Y. Zhao and P. Tsiotras, “Density functions for mesh refinement in numerical optimal control,” *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 1, pp. 271–277, 2011.
- [36] M. Li and W. Zhao, “Visiting power laws in cyber-physical networking systems,” *Mathematical Problems in Engineering*, vol. 2012, Article ID 302786, 13 pages, 2012.
- [37] M. Li, “Fractal time series—a tutorial review,” *Mathematical Problems in Engineering*, vol. 2010, Article ID 157264, 26 pages, 2010.
- [38] M. Li, M. Scialia, and C. Toma, “Nonlinear time series: computations and applications,” *Mathematical Problems in Engineering*, vol. 2010, Article ID 101523, 5 pages, 2010.
- [39] M. Li and W. Zhao, “Quantitatively investigating the locally weak stationarity of modified multifractional Gaussian noise,” *Physica A*, vol. 391, no. 24, pp. 6268–6278, 2012.
- [40] M. Li, C. Cattani, and S. Y. Chen, “Viewing sea level by a one-dimensional random function with long memory,” *Mathematical Problems in Engineering*, vol. 2011, Article ID 654284, 13 pages, 2011.
- [41] M. Li, “Approximating ideal filters by systems of fractional order,” *Computational and Mathematical Methods in Medicine*, vol. 2012, Article ID 365054, 6 pages, 2012.
- [42] C. Cattani, E. Laserra, and I. Bochicchio, “Simplicial approach to fractal structures,” *Mathematical Problems in Engineering*, vol. 2012, Article ID 958101, 21 pages, 2012.
- [43] A. V. Rao, D. A. Benson, and C. L. Darby, “User’s manual for GPOPS: a Matlab software for solving optimal control problems using pseudospectral methods,” 2011, <http://www.gpops.org>.
- [44] P. E. Gill, W. Murray, and M. A. Saunders, “SNOPT: an SQP algorithm for large-scale constrained optimization,” *SIAM Review*, vol. 47, no. 1, pp. 99–131, 2005.
- [45] K. Holmström, A. O. Göran, and M. M. Edvall, *Users Guide For TOMLAB /SNOPT*, Mälardalen University, Department of Mathematics and Physics, Västerås, Sweden, 2006.
- [46] A. V. Rao, “Application of a dichotomic basis method to performance optimization of supersonic aircraft,” *Journal of Guidance, Control, and Dynamics*, vol. 23, no. 3, pp. 570–573, 2000.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

