AIAA Guidance, Navigation, and Control Conference
2 - 5 August 2010, Toronto, Ontario Canada

AIAA 2010-8272

# A Preliminary Analysis of a Variable-Order Approach to Solving Optimal Control Problems Using Pseudospectral Methods

Christopher L. Darby*
William W. Hager†
Anil V. Rao‡

*University of Florida*
*Gainesville, FL 32611*

**An iterative variable-order $hp-$adaptive pseudospectral method is presented for solving optimal control problems. In the method of this paper, the accuracy of the solution in each segment is assessed by computing the violations in the discretized differential-algebraic equations at a large number of sample points in each segment. The accuracy on the subsequent mesh is then improved by redistributing the collocation points using a functional that depends upon the curvature of the trajectory. The algorithm iterates on the collocation point distribution until the accuracy criterion is satisfied to a user-specified tolerance. The method allows for different degree polynomial approximations in each segment, but the algorithm is designed in such a manner that the maximum polynomial degree in every segment remains small. It is found that a computationally efficient algorithm results from the ability to vary the degree of the polynomial in each segment while keeping the maximum degree bounded. Finally, an example is considered to demonstrate the effectiveness of the method.**

## I. Introduction

In recent years, the class of *pseudospectral methods* for solving optimal control problems has increased in popularity.[1–8] In a pseudospectral method, the collocation points are chosen based on accurate quadrature rules and the basis functions are typically Chebyshev or Lagrange polynomials. Pseudospectral methods can be employed in three different forms. In an $h-$method (also called a *local collocation method*), the state is approximated using many fixed- and low-degree segments. Convergence in an $h-$method is then achieved by increasing the number of segments. In a $p-$method (also called a *global collocation method*), a single segment is used and convergence is achieved by increasing the degree of the polynomial. The most well developed pseudospectral methods are of the $p-$type and are the *Lobatto pseudospectral method*,[1] the *Gauss pseudospectral method*,[2,3] and the *Radau pseudospectral method*.[6–8]

[*] PhD Candidate. Department of Mechanical and Aerospace Engineering. E-mail: cdarby@ufl.edu

[†] Professor. Department of Mathematics, E-mail: hager@ufl.edu

[‡] Assistant Professor. Department of Mechanical and Aerospace Engineering. E-mail: anilvrao@ufl.edu. Corresponding Author.

The literature on pseudospectral methods has focused largely on $p-$methods.[1–3,7,8] As it turns out, however, relying on convergence using a $p-$method has limitations. For even many smooth problems it may be possible to obtain an accurate approximation only when a fairly large-degree global polynomial is used. In addition, the convergence rate of a $p-$method on a problem whose formulation or solution is nonsmooth may be extremely slow, resulting in a poor approximation even when a high degree polynomial is used. A second limitation of a $p-$method is that the use of a high-degree global polynomial results in an NLP whose density grows quickly as a function of the number of NLP variables due to the dense blocks that arise from the global differentiation matrix. Thus, a $p-$method becomes computationally intractable for problems when an excessively large-degree polynomial is required. An alternative to a $p-$method, as mentioned above, is an $h-$method.[6,9,10] While $h-$methods are computationally more tractable than $p-$methods, they may require that a large number of mesh intervals be used in order to achieve an acceptable accuracy. Furthermore, because exponential convergence is lost when using an $h-$ method, achieving a given accuracy may result in an extremely large NLP. If the NLP is sufficiently large, it may be difficult to solve in a computationally efficient manner.

In order to increase the utility of pseudospectral methods, in this paper we combine the best features of both an $h-$method and a $p-$method to form a so called *$hp-$method.* As its name implies, an $hp-$method is one where the number of segments, the segment widths, and the polynomial degree in each segment is determined algorithmically. Previously, $hp-$methods have been developed in the context of finite elements in mechanics and spectral methods in fluid dynamics. In particular, Refs. 11–15 describe the mathematical properties of $h$, $p$, and $hp$ methods for finite elements. Ref. 16 showed the application of an $hp-$adaptive least-squares spectral element method (LS-SEM) for solving hyperbolic partial differential equations. Ref. 17 developed an adaptive spectral least-squares collocation scheme for Burgers' equation. Ref. 18 showed the use of an $hp-$adaptive LS-SEM for solving the population balance equation, while Ref. 19 developed an $hp-$adaptive spectral element solver for reactor modeling. Finally, an overview of $hp-$adaptive spectral element methods for solving problems in computational fluid dynamics can be found in Ref. 20.

In this paper we develop a so called $hp-$adaptive method for solving optimal control problems. In the method developed in this paper, the accuracy of the solution is improved either by increasing the degree of the polynomial or increasing the number of low-degree segments. The distribution of collocation points is determined using the integral of a density function that depends upon the curvature in each segment. If the density function in a particular segment is large at particular points in a segment relative to other points in the segment, then the accuracy is increased by subdividing the segment and using low-degree polynomials in the resulting subdivision. Otherwise, accuracy is improved by increasing the degree of approximating polynomial in the segment. The method is demonstrated on an example and is found to be a viable method for efficiently and accurately solving complex optimal control problems using pseudospectral methods.

It is noted that the method of this paper shares one similarity to the work of Ref. 21. Specifically, in the method of Ref. 21, the mesh is refined globally based on the integral of a curvature density function, and the density of collocation points is proportional to the magnitude of the integrated density function. The method presented here also uses curvature, but differs in many ways from the method of Ref. 21. First, if a segment needs to be subdivided, the integral of the curvature function is used within each segment to determine the locations of the new subintervals. As a result, segments changes are much more local in the method of this paper as compared with the method of Ref. 21. Furthermore, the algorithm of this paper allows for a different degree polynomial approximation in each segment. Finally, the growth rate of the mesh in this algorithm is a function of the error in the current solution and the desired user prescribed accuracy.

This paper is organized as follows. In Section II we define the Bolza optimal control problem. In Section III we formulate the Radau pseudospectral method, which is the basis for the method presented in this paper. In Section IV we present our variable-order adaptive pseudospectral method. In Section V we provide an application of the method. Finally, in Section VI we provide conclusions.

## II. Bolza Optimal Control Problem

Consider the following fairly general optimal control problem in Bolza form. Minimize the cost functional

$$J = \mathbf{\Phi}(\mathbf{x}(t_0), t_0, \mathbf{x}(t_0), t_f) + \int_{t_0}^{t_f} g(\mathbf{x}(t), \mathbf{u}(t), t)\, dt \tag{1}$$

subject to the dynamic constraints

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t), \tag{2}$$

the inequality path constraints

$$\mathbf{C}(\mathbf{x}(t), \mathbf{u}(t), t) \leq \mathbf{0}, \tag{3}$$

and the boundary conditions (i.e., the event constraints)

$$\boldsymbol{\phi}(\mathbf{x}(t_0), t_0, \mathbf{x}(t_0), t_f) = \mathbf{0}, \tag{4}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the state, $\mathbf{u}(t) \in \mathbb{R}^m$ is the control, and $t$ is time. For convenience all quantities will be treated as *row vectors*, that is, if $\mathbf{y}(t) \in \mathbb{R}^q$ is a vector function of $t$, then $\mathbf{y}(t) \equiv \left[\begin{array}{ccc} y_1(t) & \cdots & y_q(t) \end{array}\right]$.

Suppose now that we divide the time interval $t \in [t_0, t_f]$ into $K$ segments $[t_{k-1}, t_k]$, $(k = 1, \ldots, K)$, where $t_0 < t_1 < t_2 < \cdots < t_K = t_f$. In addition, suppose in each segment $k \in [1, \ldots, K]$ the variable $t \in [t_{k-1}, t_k]$ is transformed to the variable $\tau \in [-1, +1]$ via the affine transformation

$$\tau = \frac{2t - (t_k + t_{k-1})}{t_k - t_{k-1}} \quad (t_{k-1} < t_k), \tag{5}$$

where

$$\frac{d\tau}{dt} = \frac{2}{t_k - t_{k-1}}, \quad (k = 1, \ldots, K) \tag{6}$$

in each segment. Next, let $\mathbf{x}^{(k)}(\tau)$ and $\mathbf{u}^{(k)}(\tau)$ be the state and control in segment $k$ in terms of the variable $\tau \in [-1, +1]$. Using Eq. (6), the Bolza optimal control problem of Eqs. (1)–(4) can be written as follows. First, the cost functional of Eq. (1) can be written as

$$J = \Phi(\mathbf{x}^{(1)}(-1), t_0, \mathbf{x}^{(K)}(+1), t_f) + \sum_{k=1}^{K} \frac{t_k - t_{k-1}}{2} \int_{-1}^{+1} g(\mathbf{x}^{(k)}(\tau), \mathbf{u}^{(k)}(\tau), \tau; t_{k-1}, t_k)\, d\tau. \tag{7}$$

Next, the dynamic constraints of Eq. (2) and the the path constraints of (3) are given in terms of $\tau$ in segment $k \in [1, \ldots, K]$, respectively, as

$$\begin{aligned} \frac{d\mathbf{x}^{(k)}(\tau)}{d\tau} &= \frac{t_k - t_{k-1}}{2}\mathbf{f}(\mathbf{x}^{(k)}(\tau), \mathbf{u}^{(k)}(\tau), \tau; t_{k-1}, t_k) && (8) \\ \mathbf{C}(\mathbf{x}^{(k)}(\tau), \mathbf{u}^{(k)}(\tau), \tau; t_{k-1}, t_k) &\leq \mathbf{0}. && (9) \end{aligned}$$

Furthermore, the boundary conditions of Eq. (4) are given as

$$\boldsymbol{\phi}(\mathbf{x}^{(1)}(-1), t_0, \mathbf{x}^{(K)}(+1), t_f). \tag{10}$$

Finally, due to the division of the problem into multiple segments, it is required that the state be continuous at each segment interface, that is, $\mathbf{x}(t_k^-) = \mathbf{x}(t_k^+)$, $(k = 1, \ldots, K-1)$.

## III. Radau Pseudospectral Method

The *hp*−method developed in this paper is developed by discretizing the multiple-segment form of the Bolza optimal control problem given in Section II using the *Radau pseudospectral method.*[7] First, within each segment $k \in [1, \ldots, K]$, the state is approximated using a polynomial of degree $N_k$,

$$\mathbf{x}^{(k)}(\tau) \approx \mathbf{X}^{(k)}(\tau) = \sum_{j=1}^{N_k+1} \mathbf{X}_j^{(k)} L_j^{(k)}(\tau), \quad L_j^{(k)}(\tau) = \prod_{\substack{l=1 \\ l \neq j}}^{N_k+1} \frac{\tau - \tau_l^{(k)}}{\tau_j - \tau_l^{(k)}}, \tag{11}$$

where $\tau \in [-1, +1]$, $L_j^{(k)}(\tau),\ (j = 1, \ldots, N_k+1)$ is a basis of Lagrange polynomials, $\left(\tau_1^{(k)}, \ldots, \tau_{N_k}^{(k)}\right)$ are the Legendre-Gauss-Radau[22] (LGR) collocation points in segment $k$, and $\tau_{N_k+1}^{(k)} = +1$ is a noncollocated point. Differentiating $\mathbf{X}^{(k)}(\tau)$ in Eq. (11) with respect to $\tau$ in segment $k$, we obtain

$$\frac{d\mathbf{X}^{(k)}(\tau)}{d\tau} \equiv \dot{\mathbf{X}}^{(k)}(\tau) = \sum_{j=1}^{N_k+1} \mathbf{X}_j^{(k)} \dot{L}_j^{(k)}(\tau). \tag{12}$$

The cost functional of Eq. (7) is then approximated using a multiple-segment LGR quadrature as

$$J \approx \Phi(\mathbf{X}_1^{(1)}, t_0, \mathbf{X}_{N_K+1}^{(K)}, t_K) + \sum_{k=1}^{K} \sum_{j=1}^{N_k} \frac{t_k - t_{k-1}}{2} w_j^{(k)} g(\mathbf{X}_j^{(k)}, \mathbf{U}_j^{(k)}, \tau_j^{(k)}; t_{k-1}, t_k), \tag{13}$$

where $\mathbf{U}_i^{(k)}$, $(i = 1, \ldots, N_k)$, are the approximations of the control at the $N_k$ LGR points in segment $k$, $\mathbf{X}_1^{(1)}$ is the approximation of $\mathbf{x}(t_0)$, and $\mathbf{X}_{N_K+1}^{(K)}$ is the approximations of $\mathbf{x}(t_f)$. Next, the continuous-time dynamics of Eq. (2) are collocated at the $N_k$ LGR points as

$$\sum_{j=1}^{N_k+1} \mathbf{X}_j^{(k)} D_{ij}^{(k)} - \frac{t_k - t_{k-1}}{2} \mathbf{f}(\mathbf{X}_i^{(k)}, \mathbf{U}_i^{(k)}, \tau_i^{(k)}; t_{k-1}, t_k) = \mathbf{0}, \quad (i = 1, \ldots, N_k), \tag{14}$$

where

$$D_{ij}^{(k)} = \dot{L}_j^{(k)}(\tau_i), \quad \left\{ \begin{array}{c} (i = 1, \ldots, N_k) \\ (i = 1, \ldots, N_k + 1) \\ (k = 1, \ldots, K) \end{array} \right. , \tag{15}$$

is the $N_k \times (N_k + 1)$ *Radau pseudospectral differentiation matrix*[7,8] in segment $k$. Furthermore, the path constraints of Eq. (3) in segment $k \in [1, \ldots, K]$ are enforced at the $N_k$ LGR points as

$$\mathbf{C}^{(k)}(\mathbf{X}_i^{(k)}, \mathbf{U}_i^{(k)}, \tau_i; t_{k-1}, t_k) \leq \mathbf{0}, \quad (i = 1, \ldots, N_k). \tag{16}$$

Finally, the event constraints are approximated as

$$\boldsymbol{\phi}(\mathbf{X}_1^{(1)}, t_0, \mathbf{X}_{N_K+1}^{(K)}, t_K) \quad = \quad \mathbf{0}. \tag{17}$$

The NLP that arises from the Radau pseudospectral approximation is then to minimize the cost function of Eq. (13) subject to the algebraic constraints of Eqs. (14)–(17). Just as in the continuous-time optimal control problem of Eqs. (7)–(10), in the Radau pseudospectral discretization it is required that the state at the end of any segment be the same as the state at the start of the next segment. Continuity in the discretized state is maintained by using the same NLP variable for the state approximation at the end of a segment as is used at the start of the next segment. In other words, if $\mathbf{X}_{N_k+1}^{(k)}$ and $\mathbf{X}_1^{(k+1)}$ are the approximations to the state at the end of segment $k$ and the start of segment $k+1$, respectively, then a single NLP variable is used for both quantities.

It is useful to observe a few key properties of the Radau pseudospectral method. First, when used as a single-segment $p-$method (see Fig. 1), the state is approximated at the $N_k$ collocation points plus the terminal point, while the control is approximated at only the $N_k$ collocation points. Replicating the single-segment structure to form a multiple-segment method, the Radau pseudospectral method has the feature that the control is approximated at every segment interface (see Fig. 2). Thus, in multiple-segment form, the control is approximated at every point at which the state is approximated with the single exception of the final point $t = t_f$. Finally, as has been shown in Refs. 7 and 8 that the differential form of the Radau pseudospectral method (that is, the form given here) can be written equivalently in *implicit integral form*. We choose to use the differential form of the method because it is the more computationally efficient of the two forms.

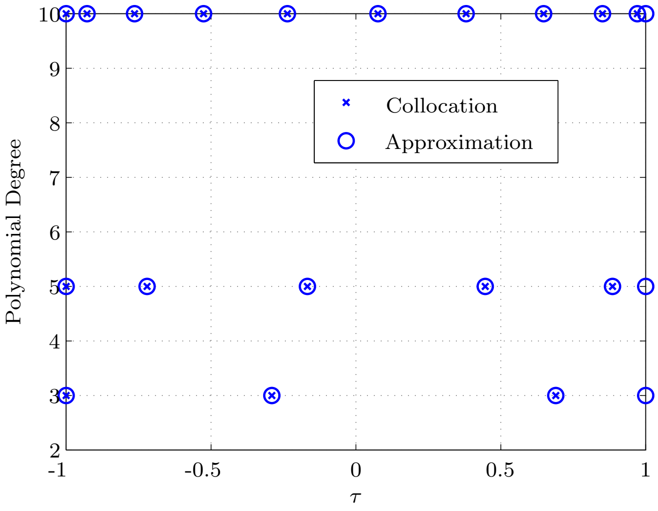


Figure 1: Radau Pseudospectral $p-$Method Collocation and Approximation Points.

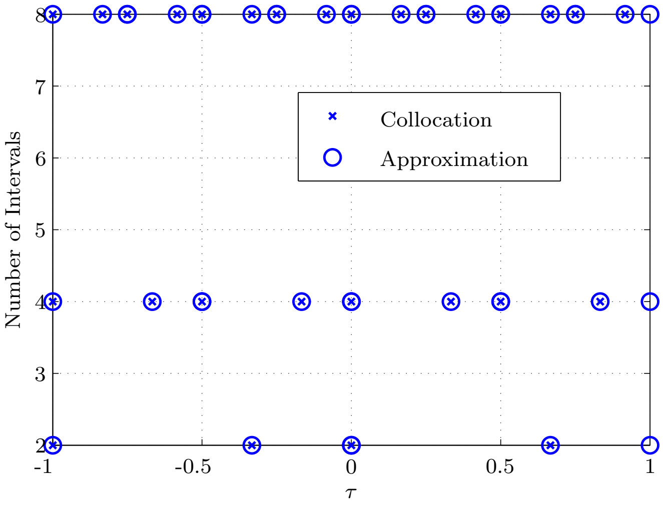


Figure 2: Radau Pseudospectral $h-$method Collocation and Approximation Points.

# IV. Variable-Order $hp-$Adaptive Algorithm

## A. Assessment of Approximation Error in a Segment

Consider again a Bolza optimal control problem on the time interval $t \in [t_0, t_f]$, where the problem has been divided into $K$ segments, the time interval segment $k \in [1, \ldots, K]$ is $[t_{k-1}, t_k]$, $N_k$ is the number of collocation points in segment $k$, and $t_K \equiv t_f$. The goal of the $hp-$adaptive algorithm is to improve the accuracy of the solution in a computationally efficient manner by determining if a particular segment of the current mesh has met a specified accuracy tolerance. If a segment has not met the accuracy tolerance, then the number and distribution of collocation points needs to be modified either by increasing the degree of approximating polynomial in the segment and/or further dividing the segment.

Let $\epsilon_d$ be a accuracy tolerance for the discretized differential-algebraic constraints (that is, the discretized dynamic and path constraints). A segment $k$ is considered to be within the accuracy tolerance if the maximum violation of the discretized differential-algebraic equations at a set of $L$ points between collocation points is below $\epsilon_d$. Unlike the state, which has a polynomial approximation as given in Eq. (11), the control has no such defined approximation. While any interpolating function whose value matches the approximation for the control at the $N_k$ LGR (collocation) points in segment $k$ is acceptable as far as the solution of the NLP is concerned, in this paper the following $(N_k)^{th}-$degree Lagrange polynomial is used to approximate the control in segments $(1, \ldots, K-1)$:

$$\mathbf{U}^{(k)}(\tau) = \sum_{i=1}^{N_k+1} \mathbf{U}_i^{(k)} \hat{L}_i^{(k)}(\tau), \quad \hat{L}_j^{(k)}(\tau) = \prod_{\substack{i=1\\ i\neq j}}^{N_k+1} \frac{\tau - \tau_i^{(k)}}{\tau_j - \tau_i^{(k)}}, \tag{18}$$

Because the final point of segment $K$ is not collocated, the following $(N_k-1)^{th}-$degree Lagrange polynomial is used to approximate the control in the final segment:

$$\mathbf{U}^{(k)}(\tau) = \sum_{i=1}^{N_k} \mathbf{U}_i^{(k)} \tilde{L}_i^{(k)}(\tau), \quad \tilde{L}_j^{(k)}(\tau) = \prod_{\substack{i=1\\ i\neq j}}^{N_k} \frac{\tau - \tau_i^{(k)}}{\tau_j - \tau_i^{(k)}}, \tag{19}$$

It is noted in Eq. (18) that the support points of the Lagrange basis $\hat{L}^{(k)}(\tau)$ are the $N_k$ LGR points plus the final point in segment $k \in [1, \ldots, K-1]$, whereas in Eq. (19) the support points of the Lagrange basis $\tilde{L}^{(k)}(\tau)$ are the $N_k$ LGR points. Suppose further than we defined a set of $L$ points $\left(\bar{\tau}_1^{(k)}, ..., \bar{\tau}_L^{(k)}\right) \in [-1, 1]$ in each segment $k \in [1, \ldots, K]$, where the $L$ sample points will be used to determine the accuracy of the differential-algebraic equations in a particular segment. Evaluating the differential-algebraic constraints at $\left(\bar{\tau}_1^{(k)}, ..., \bar{\tau}_L^{(k)}\right)$, we have

$$\begin{aligned} |\dot{\mathbf{X}}^{(k)}(\bar{\tau}_l^{(k)}) - \frac{t_k - t_{k-1}}{2}\mathbf{f}^{(k)}(\mathbf{X}_l^{(k)}, \mathbf{X}_l^{(k)}, \bar{\tau}_l^{(k)}; t_{k-1}, t_k)| &= \mathbf{e}_l^{(k)}, & (20)\\ \mathbf{C}_1^{(k)}(\mathbf{X}_l^{(k)}, \mathbf{U}_l^{(k)}, \bar{\tau}_l^{(k)}; t_{k-1}, t_k) &= \mathbf{b}_l^{(k)}, & (21) \end{aligned}$$

where $l = (1, .., L)$. If every element of $\mathbf{e}_l^{(k)}$ and $\mathbf{b}_l^{(k)}$, $l \in [1, \ldots, L]$, are less than $\epsilon_d$, then the current segment is considered an accurate approximation to the optimal control problem. If any element of $\mathbf{e}_l^{(k)}$ or $\mathbf{b}_l^{(k)}$ is greater than $\epsilon_d$, then the current segment is not considered accurate for this approximation and needs to be modified by either increasing the degree of the approximating polynomial or dividing the current segment into more segments.

It is noted that Eq. (21) measures the quality with which the path constraint of Eq. (3) is satisfied in segment $k$ at a point $l$. If we are in a region of the trajectory where the path constraint is inactive, $\mathbf{b}_l^{(k)}$ will be negative, and therefore, will always be less than $\epsilon_d$. If we are in a region where the path constraint is active, $\mathbf{b}_l^{(k)}$ may have positive values. If any of these positive values exceeds $\epsilon_d$, the path constraint is not satisfied to the desired accuracy tolerance.

## B. Determination of Polynomial Degree Increase or Segment Division

If the accuracy in segment $k$ needs to be improved, the first step is to determine if this segment will be modified by increasing the degree of the approximation, $N_k$, or by subdividing the segment. Suppose that $X_m^{(k)}(\tau)$ is the component of the state approximation in segment $k$ that corresponds to the maximum value of either $\mathbf{e}_l^{(k)}$ or $\mathbf{b}_l^{(k)}$, $l \in [1, \ldots, L]$, and let this maximum error be denoted by $e_{\max}^{(k)}$. Next, the curvature of the $m^{th}$ component of the state in segment $k$ is computed as

$$\kappa^{(k)}(\tau) = \frac{|\ddot{X}_m^{(k)}(\tau)|}{\left|\left[1 + \dot{X}_m^{(k)}(\tau)^2\right]^{3/2}\right|}. \tag{22}$$

Let $\kappa_{\max}^{(k)}$ and $\bar{\kappa}^{(k)}$ be the maximum and mean value of $\kappa^{(k)}(\tau)$, respectively, over all components of the state in segment $k$. Finally, let

$$r_k = \frac{\kappa_{\max}^{(k)}}{\bar{\kappa}^{(k)}}. \tag{23}$$

If $r_k < r_{\max}$, the curvature is considered uniform in this segment and a larger degree polynomial is used to obtain a better approximation to segment $k$. If $r_k > r_{\max}$, then the curvature is large relative to the rest of the segment and the segment is divided into more segments.

## C. Determination of New Polynomial Degree within a Segment

Let $N_k^-$ and $N_k^+$ denote the number of collocation points in a segment before and after an update of the polynomial degree in a segment, $N_k^+$ is the following function of $N_k^-$, the desired error tolerance, $\epsilon_d$, and the maximum violation of our dynamic collocation or path constraint equations, $e_{\max}^{(k)}$:

$$N_k^+ = N_k^- + \mathbf{ceil}(\log_{10}(e_{\max}^{(k)}) - \log_{10}(\epsilon_d)) + X \tag{24}$$

where $X$ is an arbitrary integer constant described in Section F and **ceil** is the operator that rounds to the next larger integer.

## D. Number and Placement of New Segments

Suppose the result of Section B is that segment $k$ needs to be divided into more segments. The following procedure is then used to determine the number of subdivisions. The number of subdivisions of segment $k$, denoted $n_k$, is given as

$$n_k = Y\mathbf{ceil}(\log_{10} e_{\max}^{(k)} - \log_{10} \epsilon_d) \tag{25}$$

where $Y$ is an arbitrary integer constant described in Section F and **ceil** is the operator that rounds the argument toward $+\infty$. The locations of the segment breaks for the new segments are then determined using a curvature density function in a manner similar to that given in Ref. 23. Specifically, let $\rho(\tau)$ be a density function,[23]

$$\rho(\tau) = c\kappa(\tau)^{1/3} \tag{26}$$

where $c$ is a constant. Next, define $F(\tau)$ to be the cumulative distribution function,

$$F(\tau) = \int_{-1}^{\tau} \rho(\zeta) d\zeta, \tag{27}$$

where $c$ is chosen such that $F(+1) = 1$. It has been shown in Ref. 23 that placement of mesh points by $\rho$ results in the best linear piecewise approximation to a curve. For $i = (1, 2, \ldots, n_k)$, with the $i^{th}$ subdivision beginning at $\tau_{i-1}$, the $(i+1)^{th}$ subdivision starts at $F(\tau_i) - F(\tau_{i-1}) = 1/n_k$. In this manner, the density of the subdivision of segment $k$ is proportional to $F(\tau)$.

### E. $hp-$Adaptive Algorithm

We now describe our $hp-$adaptive algorithm that uses the components described in Sections IV–A through IV–D. First, let $m$ be the minimum allowable polynomial degree (which is equal to the number of LGR collocation points) in any segment. Next, let $K$ be the number of segments used on any grid iteration. Furthermore, let $m$ be the polynomial degree chosen in each segment on the initial grid. Generally, the initial value of $K$ should be large enough so that both a solution can be obtained on the first grid and the algorithm can iterate to find solutions on subsequent grids, but should be small enough that a solution on the first grid is obtained in a computationally efficient manner. The initial polynomial degree $m$ should be small (e.g., two, three, or four collocation points) so that the NLP is as sparse as possible on the first grid iteration. Once an initial grid of segments and collocation points within each segment have been chosen, the $hp-$adaptive algorithm proceeds as described below.

1. If this is not the first grid iteration, compare the current state solution to the previous state solution. If the maximum relative difference between these two state solutions is less than $\epsilon$, then stop. Otherwise, continue to Step 2.

**Begin: For $i = 1, \ldots, K$,**

2. Assess violations in the discretized differential-algebraic equations evaluated at a uniformly distributed set of $L$ points between collocation points in each segment. If the error in the collocation equations or path constraints at the points sampled between the collocation points are satisfied to the tolerance $\epsilon_d$, leave the segment unchanged. Otherwise, continue to Step 3. If the violations in the discretized differential-algebraic equations is less than $\epsilon_d$ in every segment, then stop.

3. If $N_k = m$ in any segment, compute the curvature and determine the value of $r_k$. If either $r_k \geq r_{\max}$ or $N_k \neq m$, proceed to Step 5. If $r_k < r_{\max}$, proceed to Step 4.

4. Determine the new number collocation points, $N_k$, in each segment by increasing $N_k$ using Eq. (24).

5. Determine the location and number of segment breaks and set $N_k = m$ in each newly created segment.

**End: For $i = 1, \ldots, K$.**

6. Return to Step 1.

### F. Remarks

The goal of the $hp-$adaptive algorithm of this paper is to use as many low-degree segments as possible and only to use larger-degree polynomials in regions where the the solution is smooth. Even for subdomains of the trajectory where larger-degree polynomials may be appropriate, the degree of the polynomial approximation is still small when compared with a $p-$method. The reason for keeping the maximum degree of the polynomial approximation relatively small is threefold. First, as discussed, the number of nonzero entries in the NLP constraint Jacobian grows as the square of the number of global collocation points. Thus, if the degree of the approximating polynomial in a segment becomes too large, the NLP becomes increasingly dense and the NLP solver has to work much harder to compute a solution. Second, the difference between the number of nonzero constraint Jacobian for an $h-$method and $hp-$method is small even when the total number of collocation points is large, provided that the maximum polynomial degree in any segment is small. Finally, using the argument of exponential convergence for a pseudospectral method, in a segment where the solution is smooth it should be possible to obtain an accurate solution using a relatively small number of collocation points in that segment. For all of the examples studied in this paper, a $7^{th}-$ degree polynomial was the largest ever computed by the algorithm described in Section IV-E.

Next, the algorithm uses the arbitrarily chosen parameters $X$ as described in Eq. (24) that controls the growth of the number of collocation points in a segment, and $Y$ as described in Eq. (25) that controls the growth in the number of segments. It is important to understand that a trade-off exists between using large and small values of either $X$ or $Y$. If either $X$ or $Y$ is sufficiently large, the algorithm will utilize fewer grids to converge to an acceptable solution, but the size of the grids may grow quickly between grids. If either $X$ or $Y$ is small, the meshes will grow much more slowly, but the algorithm may require a much larger number of grid iterations to achieve an acceptable solution.

## V. Example

The $hp-$adaptive method of Section IV is now demonstrated on an example. The example was solved using the open-source software GPOPS[5] with the NLP solver SNOPT.[24] GPOPS was modified to include the algorithm of this paper. All computations were performed using a 2.5 GHZ Core 2 Duo Macbook Pro running Mac OS-X 10.6.3 with MATLAB R2009b. The following values were used for the parameters described in Section IV: $Y = 2$, $X = 1$, $L = 30$, $r_{\max} = 2$ and $\epsilon_d = 10\epsilon$. When comparing the $hp-$method against a $p-$method, the same stopping criterion is used. In the case of a $p-$method, if a solution is unacceptable, the degree of the global polynomial is increased by ten until a solution is achieved. When comparing the $hp-$method against an $h-$method, the parameter $r_{\max}$ is set less than one so that only segment division is allowed and the polynomial degree can never change. Finally, the terminology "$hp - x$" will be used to denote an $hp-$method with a minimum polynomial degree of $x$. Finally, in the results that follow we will employ a second-degree Radau $h-$method.

Consider the following optimal control problem.[25] Minimize the cost functional

$$J = -\phi(t_f) \tag{28}$$

subject to the dynamic constraints

$$
\begin{aligned}
\dot{r} &= v\sin\gamma, \\
\dot{\theta} &= \frac{v\cos\gamma\sin\psi}{r\cos\phi}, \\
\dot{\phi} &= \frac{v\cos\gamma\cos\psi}{r}, \\
\dot{v} &= -D - g\sin\gamma, \\
\dot{\gamma} &= \frac{L\cos\alpha - \cos\gamma(g - v^2/r)}{v}, \\
\dot{\psi} &= \frac{L\sin\alpha/\cos\gamma + v^2\cos\gamma\sin\psi\tan\phi/r}{v},
\end{aligned}
\tag{29}
$$

and the boundary conditions

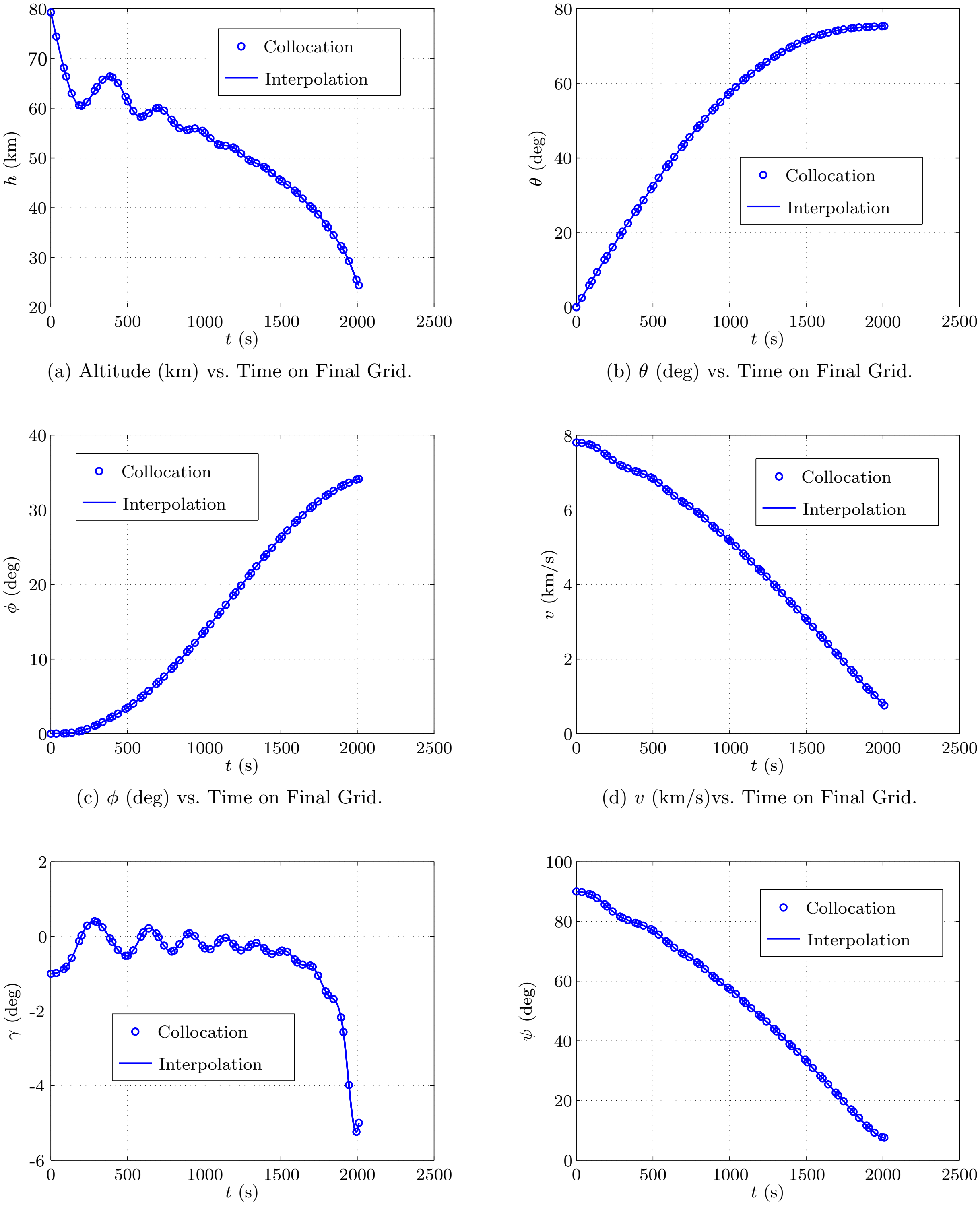
$$
\begin{array}{rclcrcl}
r(0) &=& 79248 + R_e \text{ m} &,& r(t_f) &=& 24384 + R_e \text{ m}, \\
\theta(0) &=& 0 \text{ deg} &,& \theta(t_f) &=& \text{Free}, \\
\phi(0) &=& 0 \text{ deg} &,& \phi(t_f) &=& \text{Free}, \\
v(0) &=& 7803 \text{ m/s} &,& v(t_f) &=& 762 \text{ m/s}, \\
\gamma(0) &=& -1 \text{ deg} &,& \gamma(t_f) &=& -5 \text{ deg}, \\
\psi(0) &=& 90 \text{ deg} &,& \psi(t_f) &=& \text{Free},
\end{array}
\tag{30}
$$

where $r$ is the geocentric radius, $\theta$ is the longitude, $\phi$ is the latitude, $v$ is the speed, $\gamma$ is the flight path angle, $\psi$ is the azimuth angle, $\alpha$ is the angle of attack, $\sigma$ is the bank angle, and $R_e = 6371203.92$ m is the radius of the Earth. Further details of this problem, including the aerodynamic model, can be found in Ref. 25. It is noted that, unlike either of the first two examples where either the control was discontinuous or an inequality path constraint was active, this problem has a much smoother solution.

Figs. 3a–3f shows the state obtained using the grid refinement algorithm of Section IV with an initial grid of 20 uniform segments and three collocation points in each segment. Table 1 shows results using the $p-$, $h-$, $hp-2-$, $hp-3-$ and $hp-4-$methods. Because the solution is smooth, a $p-$method should perform well. Against intuition, it was not possible to obtain a solution using the $p-$method for $\epsilon > 10^{-5}$. Next, the $h-$ and $hp-$methods outperformed the $p-$method even for the accuracy tolerance where a solution could be obtained using a $p-$method. As the accuracy tolerance is tightened, the $h-$method requires a very large number of collocation points to achieve an acceptable solution, which significantly increased the computational cost when compared with the $hp-$methods. On the other hand, all of the $hp-$methods performed better when compared with an $h$ approximation with the higher order $hp-4$ method performing the best. For $\epsilon = 10^{-7}$ and $\epsilon = 10^{-8}$, the $hp-4-$method produces a solution two orders of magnitude faster with less than half the Jacobian entries for $\epsilon = 10^{-7}$ and less than a fourth of the Jacobian entries for $\epsilon = 10^{-8}$. For the lower accuracy solutions, the $h-$ and $hp-$methods performed similarly.

| $\epsilon$ | Strategy | CPU Time (s) | Collocation Points | Segments | Grid Iterations | Jacobian Entries |
|---|---|---|---|---|---|---|
| $10^{-5}$ | $p$ | 8.47 | 30 | 1 | 2 | 7202 |
|  | $hp-4$ | 4.58 | 80 | 20 | 1 | 6722 |
|  | $hp-3$ | 3.86 | 60 | 20 | 1 | 4682 |
|  | $hp-2$ | 2.46 | 45 | 20 | 2 | 3380 |
|  | $h$ | 3.62 | 50 | 25 | 3 | 3602 |
| $10^{-6}$ | $p$ | – | – | – | – | – |
|  | $hp-4$ | 7.52 | 88 | 22 | 2 | 7394 |
|  | $hp-3$ | 11.94 | 71 | 23 | 5 | 5600 |
|  | $hp-2$ | 10.05 | 71 | 20 | 5 | 5934 |
|  | $h$ | 10.91 | 92 | 46 | 4 | 6626 |
| $10^{-7}$ | $p$ | – | – | – | – | – |
|  | $hp-4$ | 14.81 | 124 | 28 | 2 | 10850 |
|  | $hp-3$ | 23.72 | 132 | 38 | 4 | 10838 |
|  | $hp-2$ | 70.21 | 173 | 44 | 6 | 14696 |
|  | $h$ | 150.44 | 334 | 167 | 5 | 24050 |
| $10^{-8}$ | $p$ | – | – | – | – | – |
|  | $hp-4$ | 41.99 | 188 | 39 | 3 | 17054 |
|  | $hp-3$ | 115.89 | 229 | 61 | 5 | 19388 |
|  | $hp-2$ | 950.62 | 483 | 121 | 14 | 41192 |
|  | $h$ | 4078.10 | 1012 | 506 | 8 | 72866 |

Table 1: Summary of Accuracy and Speed for Example Problem Using Various Collocation Strategies and Accuracy Tolerances.



(a) Altitude (km) vs. Time on Final Grid.

(b) $\theta$ (deg) vs. Time on Final Grid.

(c) $\phi$ (deg) vs. Time on Final Grid.

(d) $v$ (km/s)vs. Time on Final Grid.

(e) $\gamma$ (deg) vs. Time on Final Grid.

(f) $\psi$ (deg) vs. Time on Final Grid.

Figure 3: State on Final Grid for Example Problem for the $hp-3$, $\epsilon = 10^{-5}$ problem.

# VI. Conclusions

An $hp-$adaptive control algorithm has been developed for solving optimal control problems using pseudospectral methods. The method of this paper adjusts the mesh size either by increasing the number of segments or by increasing the degree of the polynomial approximation in each segment. The integral of a density function of curvature is used to determine the distribution of collocation points in each segment on the next mesh. The approach is demonstrated on a smooth example. When compared against global pseudospectral methods, it is found that the approach of this paper produces more accurate solutions in a more computationally efficient manner. Furthermore, the method of this paper produced comparable or better solutions when compared with fixed-degree local collocation methods.

# References

[1] Elnagar, G., Kazemi, M., and Razzaghi, M., "The Pseudospectral Legendre Method for Discretizing Optimal Control Problems," *IEEE Transactions on Automatic Control*, Vol. 40, No. 10, 1995, pp. 1793–1796.

[2] Benson, D. A., *A Gauss Pseudospectral Transcription for Optimal Control*, Ph.D. thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2004.

[3] Benson, D. A., Huntington, G. T., Thorvaldsen, T. P., and Rao, A. V., "Direct Trajectory Optimization and Costate Estimation via an Orthogonal Collocation Method," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 6, November-December 2006, pp. 1435–1440.

[4] Huntington, G. T., *Advancement and Analysis of a Gauss Pseudospectral Transcription for Optimal Control*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2007.

[5] Rao, A. V., Benson, D. A., Darby, C. L., Francolin, C., Patterson, M. A., Sanders, I., and Huntington, G. T., "Algorithm: GPOPS, A Matlab Software for Solving Multiple-Phase Optimal Control Problems Using the Gauss Pseudospectral Method," *ACM Transactions on Mathematical Software*, Vol. 37, No. 2, April–June 2010, pp. 22:1–22:39.

[6] Kameswaran, S. and Biegler, L. T., "Convergence Rates for Direct Transcription of Optimal Control Problems Using Collocation at Radau Points," *Computational Optimization and Applications*, Vol. 41, No. 1, 2008, pp. 81–126.

[7] Garg, D., Patterson, M. A., Darby, C. L., Francolin, C., Huntington, G. T., Hager, W. W., and Rao, A. V., "Direct Trajectory Optimization and Costate Estimation of Finite-Horizon and Infinite-Horizon Optimal Control Problems via a Radau Pseudospectral Method," *Computational Optimization and Applications*, Published Online: October 6, 2009. DOI 10.1007/s10589-009-9291-0. http://www.springerlink.com/content/n851q6n343p9k60k/.

[8] Garg, D., Patterson, M. A., Hager, W. W., Rao, A. V., Benson, D. A., and Huntington, G. T., "A Unified Framework for the Numerical Solution of Optimal Control Problems Using Pseudospectral Methods," *Automatica*, Accepted for Publication, June 2010.

[9] Cuthrell, J. E. and Biegler, L. T., "On the Optimization of Differential-Algebraic Processes," *AIChe Journal*, Vol. 33, No. 8, August 1987, pp. 1257–1270.

[10] Cuthrell, J. E. and Biegler, L. T., "Simultaneous Optimization and Solution Methods for Batch Reactor Control Profiles," *Computers and Chemical Engineering*, Vol. 13, No. 1/2, 1989, pp. 49–62.

[11] Babuska, I. and Suri, M., "The $p$ and $hp$ Version of the Finite Element Method, an Overview," *Computer Methods in Applied Mechanics and Engineering*, Vol. 80, 1990, pp. 5–26.

[12] Babuska, I. and Suri, M., "The $p$ and $hp$ Version of the Finite Element Method, Basic Principles and Properties," *SIAM Review*, Vol. 36, 1994, pp. 578–632.

[13] Gui, W. and Babuska, I., "The $h$, $p$, and $hp$ Versions of the Finite Element Method in 1 Dimension. Part I. The Error Analysis of the $p$ Version," *Numerische Mathematik*, Vol. 49, 1986, pp. 577–612.

[14] Gui, W. and Babuska, I., "The $h$, $p$, and $hp$ Versions of the Finite Element Method in 1 Dimension. Part II. The Error Analysis of the $h$ and $h-p$ Versions," *Numerische Mathematik*, Vol. 49, 1986, pp. 613–657.

[15] Gui, W. and Babuska, I., "The $h$, $p$, and $hp$ Versions of the Finite Element Method in 1 Dimension. Part III. The Adaptive $h-p$ Version," *Numerische Mathematik*, Vol. 49, 1986, pp. 659–683.

[16] Galvao, A., Gerritsma, M., and Maerschalk, B. D., "*hp*-Adaptive Least-Squares Spectral Element Method for Hyperbolic Partial Differential Equations," *Journal of Computational and Applied Mathematics*, Vol. 215, No. 2, June 2008, pp. 409–418.

[17] Heinrichs, W., "An Adaptive least Squares Scheme for the Burgers Equation," *Numerical Algorithms*, Vol. 44, No. 1, January 2007, pp. 1–10.

[18] Dorao, C. A. and Jakobsen, H. A., "*hp*-Adaptive Least Squares Spectral Element Method for Population Balance Equations," *Applied Numerical Mathematics*, Vol. 58, No. 5, May 2008, pp. 563–576.

[19] Dorao, C. A., Fernandino, M., Jakobsen, H. A., and Svendsen, H. F., "*hp*-Adaptive Spectral Element Solver for Reactor Modeling," *Chemical Engineering Science*, Vol. 64, No. 5, March 2009, pp. 904–911.

[20] Karniadakis, G. and Sherwin, S., *Spectral/hp Element Methods for CFD*, Oxford University Press, Oxford, UK, 1999.

[21] Zhao, Y. and Tsiotras, P., "A Density-Function Based Mesh Refinement Algorithm for Solving Optimal Control Problems," Infotech and Aerospace Conference, AIAA Paper 2009-2019, Seattle, Washington, April 2009.

[22] Abramowitz, M. and Stegun, I., *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, New York, 1965.

[23] Zhao, Y. J., "Optimal Pattern of Glider Dynamic Soaring," *Optimal Control Applications and Methods*, Vol. 25, 2004, pp. 67–89.

[24] Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Review*, Vol. 47, No. 1, January 2002, pp. 99–131.

[25] Betts, J. T., *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, SIAM Press, Philadelphia, 2010.