Adaptive Mesh Refinement Method for Optimal Control Using Decay Rates of Legendre Polynomial Coefficients

Fengjin Liu, William W. Hager, and Anil V. Rao

***Abstract*— An adaptive mesh refinement method for solving optimal control problems is described. The method employs orthogonal collocation at Legendre–Gauss–Radau points. Accuracy in the method is achieved by adjusting the number of mesh intervals, the polynomial degree within each mesh interval, and, when possible, reducing the mesh size. The decision to increase the degree of the polynomial within a mesh interval or to create new mesh intervals is based on the decay rate of the coefficients of a Legendre polynomial approximation of the state as a function of the index of the Legendre polynomial expansion. The polynomial degree in a mesh interval is increased if the Legendre polynomial coefficient decay rate exceeds a user-specified threshold. Otherwise, the mesh interval is divided into subintervals. The method developed in this brief is then demonstrated on two examples, one of which is a practical problem in aircraft performance optimization. It is found that the approach developed in this brief is more efficient, is simpler to implement, and requires the specification of fewer user-defined parameters when compared with recently developed adaptive mesh refinement methods for optimal control.**

***Index Terms*— Collocation methods, Legendre polynomials, mesh refinement, optimal control.**

I. Introduction

DIRECT collocation methods for solving a continuous optimal control problem are implicit simulation methods where the state and control are parameterized and the constraints in the continuous optimal control problem are enforced at a specially chosen set of collocation points. The approximation obtained using collocation results in a finite-dimensional nonlinear programming problem (NLP) [1] and the NLP is then solved using a well-known software [2]. Traditional direct collocation methods take the form of an h method (for example, Euler or Runge–Kutta methods), where the domain of interest is divided into a mesh, the state is approximated using the same fixed-degree polynomial in each mesh interval, convergence is achieved by increasing the number and placement of the mesh points [1].

Manuscript received October 24, 2015; revised May 3, 2016 and January 6, 2017; accepted April 21, 2017. Date of publication June 6, 2017; date of current version June 11, 2018. This work was supported in part by the U.S. Office of Naval Research under Grant N00014-15-1-2048, in part by the U.S. National Science Foundation under Grant CBET-1404767, Grant DMS-1522629, and Grant CMMI-1563225, and in part by the U.S. Air Force Research Laboratory under Contract FA8651-08-D-0108/0054. This paper was recommended by Associate Editor A. Serrani. *(Corresponding author: Anil V. Rao.)*

F. Liu is with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: cliu121@ufl.edu).

W. W. Hager is with the Department of Mathematics, University of Florida, Gainesville, FL 32611 USA (e-mail: hager@ufl.edu).

A. V. Rao is with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: anilvrao@ufl.edu).

Color versions of one or more of the figures in this paper ares available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCST.2017.2702122

In contrast to an h method, in recent years so called p methods have been developed. In a p method, the number of intervals is fixed and convergence is achieved by increasing the degree of the polynomial approximation in each interval. In order to achieve maximum effectiveness, p methods have been developed using collocation at *Gaussian quadrature* points [3]–[5]. For problems whose solutions are smooth and well-behaved, Gaussian quadrature collocation converges at an exponential rate [6]–[10]. Gauss quadrature collocation methods use either Legendre–Gauss [4], Legendre–Gauss–Radau (LGR) [5], or Legendre–Gauss–Lobatto [3] points.

Various h or p direct collocation methods have been developed previously. Gong *et al.* [11] describe a p method that used a differentiation matrix to identify potential discontinuities in the solution. Zhao and Tsiotras [12] develop an h method that uses a density function to generate a sequence of nondecreasing size meshes on which to solve the optimal control problem. Betts [1] develops an error estimate for the state using a low-order h method based on the difference between the integration of the dynamics and the integration of the time derivative of the state. Different from all of this previous research where the order of the method is fixed and the mesh can only increase in size, in the method of this brief, the degree of the polynomial approximation is varied and the mesh size can be reduced.

Although h methods have been used extensively and p methods are useful on certain types of problems, both the h and p approaches have limitations. In the case of an h method, it may be required to use an extremely fine mesh to improve accuracy. In the case of a p method, it may be required to use an unreasonably large degree polynomial to improve accuracy. In order to reduce significantly the size of the finite-dimensional approximation, and thus improve computational efficiency of solving the NLP, in recent years, the new class of hp collocation methods has been developed for solving an optimal control problem. In an hp method, both the number of mesh intervals and the degree of the approximating polynomial within each mesh interval are allowed to vary. While hp adaptive methods can be developed using classical discretizations (for example, using a Runge–Kutta method where the order of the method in a mesh interval can be varied), employing Gaussian quadrature has advantages over classical approaches. First, exponential convergence can be achieved by increasing the degree of the polynomial approximation in segments where the solution is smooth. Second, Runge phenomenon (where the error at the ends of a mesh interval becomes very large as the polynomial degree is increased) is eliminated using a Gaussian quadrature. Third, less mesh refinement needs to be performed using a Gaussian quadrature when compared with a classical method, because the mesh only needs to be refined in segments where smoothness is lost. While *hp* methods were originally developed as finite-element methods for solving partial differential equations [13], in the past few years, *hp* methods have been extended to optimal control and a convergence theory for these methods has been established [6]–[10].

In this brief, a novel *hp*-adaptive mesh refinement method for optimal control is developed. The novelty of the method described in this brief lies in the result that was proven in [14]. Specifically, [14] proved that the error in a Legendre polynomial approximation of an analytic function decays as a function of the polynomial degree at the same rate as the Legendre polynomial coefficients used in the approximation (it is noted that the result of [14] was introduced in [15], while an *hp* method for solving elliptic partial differential equations was developed in [16] based on the result of [15]). The *hp* mesh refinement method of this brief is then developed using the following approach that employs the rigorously proven result of [14]. First, using the result in [14], the decay rate of the state within a mesh interval is estimated to be the decay rate of the Legendre polynomial coefficients within the mesh interval. If the decay rate within the mesh interval is larger than a user-specified threshold decay rate, then the state in the mesh interval is assumed to be an analytic function and the degree of the approximating polynomial is increased. If, on the other hand, the decay rate within the mesh interval is smaller than the user-specified threshold decay rate, then the state within the mesh interval is assumed to be a non-analytic function and the mesh interval is divided into subintervals. The novelty of the method described in this brief lies in the fact that it enables utilizing the exponential convergence of a Gauss quadrature when the state is estimated to be an analytic function and creates new mesh intervals only when the state is estimated to be non-analytic. Moreover, the method is straightforward to implement, making it usable in practical optimal control problems. To demonstrate its utility, the method is applied to two examples where the second example is a nontrivial minimum-time supersonic aircraft climb optimal control problem. It is found that the approach developed in this brief is more efficient, is simpler to implement, and requires the specification of fewer user-defined parameters when compared with recently developed adaptive mesh refinement methods for optimal control.

This brief is organized as follows. Section II describes the basis of the new *hp*-adaptive method based on the decay rate of the Legendre polynomial coefficients. Section III describes the optimal control problem in Bolza form, while Section IV describes the LGR collocation method. Section V describes the new *hp*-adaptive method developed in this brief based on the approach developed in Section II. Section VI provides the steps that comprise the *hp*-adaptive algorithm based on the components described in Section V. Section VII demonstrates the *hp*-adaptive method on two examples from the open literature and compares the results obtained with this method with the results obtained using the *hp*-adaptive methods in [17] and [18]. Section VIII provides a discussion of the results. Finally, Section IX provides conclusions on this research.

II. Foundation of *hp*-Adaptive Mesh Refinement Method

In this section, we provide the basis for the new *hp*-adaptive mesh refinement method described in Section V. The new *hp*-adaptive method developed in this brief is motivated by the comparison study of *hp*-adaptive methods for elliptic partial differential equations. In particular, in [14], it has been shown that the decay rate of a Legendre polynomial approximation of a piecewise smooth function can be estimated from the coefficients of the Legendre polynomial approximation. The relationship established in [14] provides a way to determine if the function being approximated is smooth or nonsmooth. If the coefficients decay sufficiently fast as the polynomial degree is increased, the function is regarded as being smooth and the approximation error is most effectively decreased by increasing the degree of the polynomial approximation. On the other hand, if the decay rate of the Legendre polynomial coefficients is sufficiently slow, the function is estimated to be nonsmooth and the error in the approximation is most effectively decreased by dividing the interval into subintervals and using a piecewise polynomial approximation. In the remainder of this section, the decay rate of the error in a Legendre polynomial approximation is derived.

A. Legendre Polynomial Approximation of Functions

Let $y(\tau)$ be a piecewise smooth bounded function on the interval $\tau \in [-1, +1]$. Suppose that it is desired to approximate $y(\tau)$ with a polynomial $Y(\tau)$ of degree N. Arbitrarily, the polynomial approximation can be expressed in terms of a basis of $N+1$ Legendre polynomials $P_i(\tau)$, $(i = 0, \ldots, N)$ as

$$y(\tau) \approx Y(\tau) = \sum_{i=0}^{N} \hat{a}_i P_i(\tau). \tag{1}$$

In [15], it has been discussed that the decay rate of the coefficients $\hat{a}_i$, $(i = 0, \ldots, N)$ can be used to estimate the smoothness of the function $y(\tau)$. In this section, it is shown that the decay rate of the Legendre coefficients $\hat{a}_i$, $(i = 0, \ldots, N)$ given in (1) is the same as the decay rate of the upper bound on the error.

Because $y(\tau)$ is a piecewise smooth bounded function, it can be represented by the infinite Legendre polynomial series

$$y(\tau) = \sum_{i=0}^{\infty} a_i P_i(\tau). \tag{2}$$

Then, the absolute error in the approximation $Y(\tau)$ of (1) satisfies the inequality

$$\begin{aligned} e &= \|y(\tau) - Y(\tau)\| \\ &= \left\| \sum_{i=0}^{\infty} a_i P_i(\tau) - \sum_{i=0}^{N} \hat{a}_i P_i(\tau) \right\| \\ &= \left\| \sum_{i=N+1}^{\infty} a_i P_i(\tau) + \sum_{i=0}^{N} (a_i - \hat{a}_i) P_i(\tau) \right\| \\ &\leq \left\| \sum_{i=N+1}^{\infty} a_i P_i(\tau) \right\| + \left\| \sum_{i=0}^{N} (a_i - \hat{a}_i) P_i(\tau) \right\| \end{aligned} \tag{3}$$

where $||f|| = \langle f, f\rangle^{1/2}$ is the norm induced by the inner product

$$\langle f, g\rangle = \int_{-1}^{1} f(\tau)g(\tau)d\tau. \tag{4}$$

Suppose now that the first and second terms in the inequality of (3) are denoted as e_t and e_a, respectively, that is,

$$e_t = \left\| \sum_{i=N+1}^{\infty} a_i P_i(\tau) \right\|, \quad e_a = \left\| \sum_{i=0}^{N} (a_i - \hat{a}_i) P_i(\tau) \right\|. \tag{5}$$

The quantities e_t and e_a represent the estimates on the *truncation error* and the *aliasing error*, respectively, and are each estimated as follows. First, using the definition of the inner product in (4), the Legendre polynomials satisfy the orthogonality property

$$\langle P_m, P_n\rangle = \int_{-1}^{1} P_m(\tau)P_n(\tau)d\tau = \frac{2}{2n+1}\delta_{mn} \tag{6}$$

where δ_{mn} is the Kronecker delta function. Using (6) together with (5), e_t and e_a are given, respectively, as

$$\begin{aligned} e_t &= \left\| \sum_{i=N+1}^{\infty} a_i P_i(\tau) \right\| = \left[\sum_{i=N+1}^{\infty} \frac{2a_i^2}{2i+1} \right]^{1/2} \\ e_a &= \left\| \sum_{i=0}^{N} (a_i - \hat{a}_i) P_i(\tau) \right\| = \left[\sum_{i=0}^{N} \frac{2(a_i - \hat{a}_i)^2}{2i+1} \right]^{1/2}. \end{aligned} \tag{7}$$

It has been noted in [19] that often $a_i \approx \hat{a}_i$, in which case e_a is negligible compared with e_t. Neglecting e_a leads to $a_i = \hat{a}_i$, $(i = 0, \dots, N)$. Next, under the assumption that $y(\tau)$ is analytic in a neighborhood of the interval $\tau \in [-1, +1]$, it has been shown in [14] that the Legendre polynomial coefficient values a_i decay like $ci10^{-\tilde{\sigma}i}$, $\tilde{\sigma} > 0$. Note, however, that there exists a slightly smaller value $\sigma > 0$, such that $ci10^{-\tilde{\sigma}i} \le c10^{-\sigma i}$. Because σ differs from $\tilde{\sigma}$ only slightly and the analysis that follows is simpler and more convenient using the more conservative exponential upper bound, in this brief, the decay rate of the form $c10^{-\sigma i}$ is used. As a result, the Legendre polynomial coefficients can be approximated using an exponential least-squares fit of the form [15]

$$A_i = c10^{-\sigma i}, \quad \sigma > 0 \tag{8}$$

to estimate the coefficients $|a_i|$, $(i = 0, \dots, \infty)$ as a function of i, where σ in (8) approximates the exponential decay rate in the Legendre polynomial coefficients [14]. Moreover, estimates of $|a_i|$, $(i > N)$, can be obtained from (8) as

$$e < \left[\sum_{i=N+1}^{\infty} \frac{2}{2i+1} a_i^2 \right]^{1/2} < \left[\sum_{i=N+1}^{\infty} a_i^2 \right]^{1/2} = \frac{c10^{-\sigma(N+1)}}{\sqrt{1 - 10^{-2\sigma}}} \tag{9}$$

where we have used the fact that $\sum_{i=0}^{\infty} a_i^2$ is a geometric series with a geometric ratio $r = 10^{-2\sigma}$. The upper bound on the error e in (3), denoted $\hat{e}$, is then given as

$$\hat{e} = \frac{c10^{-\sigma(N+1)}}{\sqrt{1 - 10^{-2\sigma}}} \tag{10}$$

and is used as the error estimate. It is seen from (8) and (10) that the coefficients $|\hat{a}_i|$ decrease at the same rate as a function of i as the error decreases as a function of N. Consequently, the Legendre coefficients $\hat{a}_i$, $(i = 0, \dots, N)$ can be used to estimate the decay rate σ of the error as a function of the degree of the polynomial approximation given in (1).

B. Assessing Function Smoothness

The decay rate of the Legendre polynomial coefficients described in Section II-A provides a way to estimate if a function is smooth or nonsmooth. Based on this estimate, if σ is sufficiently large, it will be preferable to approximate the function using a single polynomial, while if σ is sufficiently small, it may be necessary to employ a piecewise polynomial approximation. Assuming a threshold of significance $\bar{\sigma}$, the function is considered to be smooth if $\sigma > \bar{\sigma}$ and is considered nonsmooth otherwise. Moreover, when $\sigma > \bar{\sigma}$, the approximation is improved by increasing the polynomial degree, while if $\sigma \le \bar{\sigma}$, the approximation is improved by dividing the time interval $\tau \in [-1, +1]$ into subintervals and using a different polynomial approximation in each subinterval. In this section, a brief study is provided that demonstrates how the decay rate of the Legendre polynomial coefficients provides an assessment of the smoothness of a function and further demonstrates the agreement of the Legendre polynomial coefficient decay rate and the upper bound of the approximation error.

C. Demonstration of Decay Rate Derived in Section II-A

In order to demonstrate the effectiveness of the error estimate derived in Section II-A, consider the following two functions on $\tau \in [-1, +1]$:

$$y_1(\tau) = \exp(\tau), \quad y_2(\tau) = \begin{cases} 1, & \tau \ge 0 \\ 0, & \tau < 0. \end{cases} \tag{11}$$

It is seen that $y_1(\tau)$ is smooth, while $y_2(\tau)$ is discontinuous. Suppose now that $y_1(\tau)$ and $y_2(\tau)$ are each approximated by a Legendre polynomial series of the form given in (2). In this brief, the Legendre polynomial coefficients are constructed by transforming a Lagrange polynomial approximation

$$y(\tau) \approx Y(\tau) = \sum_{j=1}^{N+1} Y_j \ell_j(\tau), \quad \ell_j(\tau) = \prod_{\substack{l=1 \\ l \ne j}}^{N+1} \frac{\tau - \tau_l}{\tau_j - \tau_l} \tag{12}$$

where the $N+1$ support points of the Lagrange polynomials $\ell_j(\tau)$, $(j = 1, \dots, N+1)$ are the N LGR points [5] $(\tau_1, \dots, \tau_N)$ on $[-1, +1]$ plus the point $\tau_{N+1} = +1$.[1] Using the Lagrange interpolation given in (12), the Legendre coefficients $(\hat{a}_0, \dots, \hat{a}_N)$ are obtained in terms of the Lagrange coefficients as

$$\begin{bmatrix} \hat{a}_0 \\ \hat{a}_1 \\ \vdots \\ \hat{a}_N \end{bmatrix} = \begin{bmatrix} P_0(\tau_1) & P_1(\tau_1) & \dots & P_N(\tau_1) \\ P_0(\tau_2) & P_1(\tau_2) & \dots & P_N(\tau_2) \\ \vdots & \vdots & \ddots & \vdots \\ P_0(\tau_{N+1}) & P_1(\tau_{N+1}) & \dots & P_N(\tau_{N+1}) \end{bmatrix}^{-1} \times \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_{N+1} \end{bmatrix}. \tag{13}$$

[1]It is noted from the property of Lagrange polynomials that $Y(\tau_i) = Y_i$, where τ_i is a support point of the Lagrange basis given in (12),

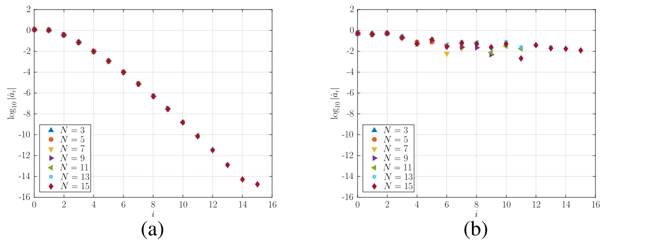


Fig. 1. Base ten logarithm of the Legendre polynomial coefficient magnitudes of polynomial approximations of functions given in (11). (a) $\log_{10}|\hat{a}_i|$ versus i of $y_1(\tau)$. (b) $\log_{10}|\hat{a}_i|$ versus i of $y_2(\tau)$.

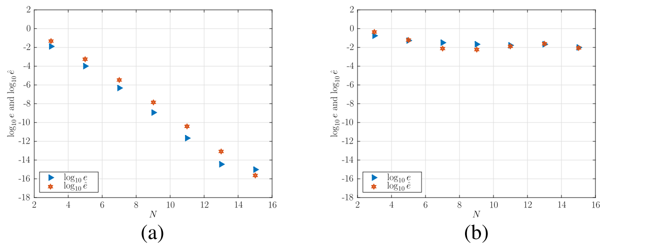


Fig. 2. Base ten logarithm of exact errors and upper bounds of error estimates of polynomial approximations of functions given in (11). (a) $\log_{10}\hat{e}$ and $\log_{10}e$ versus N of $y_1(\tau)$. (b) $\log_{10}\hat{e}$ and $\log_{10}e$ versus N of $y_2(\tau)$.

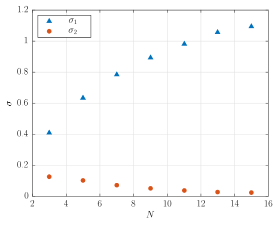


Fig. 3. Decay rates of Legendre polynomial coefficient magnitudes of polynomial approximations of functions given in (11).

The coefficients $\hat{a}_i,\ (i=0,\dots,N)$ are then used to construct a least-squares exponential fit of the form given in (8). Fig. 1 shows $\log_{10}|\hat{a}_i|$ as a function of i and Fig. 2 shows $\log_{10}\hat{e}$ and $\log_{10}e$ as a function of N for the two Legendre polynomial approximations. Several features can be seen in the results. First, Fig. 2 shows that the error estimates are in close proximity to the actual error. Second, it is seen that the error estimate of the smooth function $y_1(\tau)$ converges rapidly as a function of the polynomial degree. On the other hand, the error estimate of the discontinuous function $y_2(\tau)$ converges slowly using a single polynomial. Furthermore, Fig. 1 shows the decay rates of the Legendre polynomial coefficients for different types of functions. The Legendre coefficients of the approximation of the smooth function (that is, the Legendre coefficients of $Y_1(\tau)$) decay much more rapidly than the coefficients of approximation $Y_2(\tau)$) of the discontinuous functions. Finally, as shown in Figs. 1 and 2, the decay rate of the Legendre polynomial coefficients of any of the functions is in close proximity to the decay rate of the error in the corresponding function approximation. Specifically, the decay rates of the errors in $y_1(\tau)$ and $y_2(\tau)$ in Fig. 2 are 1.17 and 0.03 respectively, which match closely the decay rates $\sigma_1 = 1.09$ and $\sigma_2 = 0.02$ for the Legendre coefficients with $N = 15$, as shown in Fig. 3.

III. Bolza Optimal Control Problem

Without loss of generality, consider the following general optimal control problem in Bolza form. Determine the state $\mathbf{y}(\tau) \in \mathbb{R}^{n_y}$ and the control $\mathbf{u}(\tau) \in \mathbb{R}^{n_u}$ on the domain $\tau \in [-1,+1]$, the initial time t_0 and the terminal time t_f that minimize the cost functional

$$\mathcal{J} = \mathcal{M}(\mathbf{y}(-1), t_0, \mathbf{y}(+1), t_f) + \frac{t_f - t_0}{2}\int_{-1}^{+1} \mathcal{L}(\mathbf{y}(\tau), \mathbf{u}(\tau), t(\tau, t_0, t_f))\,d\tau \quad (14)$$

subject to the dynamic constraints

$$\frac{d\mathbf{y}}{d\tau} - \frac{t_f - t_0}{2}\mathbf{a}(\mathbf{y}(\tau), \mathbf{u}(\tau), t(\tau, t_0, t_f)) = \mathbf{0} \quad (15)$$

the inequality path constraints

$$\mathbf{c}_{\min} \le \mathbf{c}(\mathbf{y}(\tau), \mathbf{u}(\tau), t(\tau, t_0, t_f)) \le \mathbf{c}_{\max} \quad (16)$$

and the boundary conditions

$$\mathbf{b}_{\min} \le \mathbf{b}(\mathbf{y}(-1), t_0, \mathbf{y}(+1), t_f) \le \mathbf{b}_{\max}. \quad (17)$$

It is noted that the time interval $\tau \in [-1,+1]$ can be transformed to the time interval $t \in [t_0, t_f]$ via the affine transformation

$$t \equiv t(\tau, t_0, t_f) = \frac{t_f - t_0}{2}\tau + \frac{t_f + t_0}{2}. \quad (18)$$

In order to discretize the optimal control problem using an *hp* method, the domain $\tau \in [-1,+1]$ is partitioned into a *mesh* consisting of K *mesh intervals* $\mathcal{S}_k = [T_{k-1}, T_k],\ k = 1,\dots,K$, where $-1 = T_0 < T_1 < \dots < T_K = +1$. The mesh intervals have the property that $\bigcup_{k=1}^{K}\mathcal{S}_k = [-1,+1]$. Let $\mathbf{y}^{(k)}(\tau)$ and $\mathbf{u}^{(k)}(\tau)$ be the state and control in $\mathcal{S}_k$. Using the transformation given in (18), the Bolza optimal control problem of (14)–(17) can then be rewritten as follows. Minimize the cost functional

$$\mathcal{J} = \mathcal{M}(\mathbf{y}^{(1)}(-1), t_0, \mathbf{y}^{(K)}(+1), t_f) + \frac{t_f - t_0}{2}\sum_{k=1}^{K}\int_{T_{k-1}}^{T_k} \mathcal{L}(\mathbf{y}^{(k)}(\tau), \mathbf{u}^{(k)}(\tau), t)\,d\tau \quad (19)$$

subject to the dynamic constraints

$$\frac{d\mathbf{y}^{(k)}(\tau)}{d\tau} - \frac{t_f - t_0}{2}\mathbf{a}(\mathbf{y}^{(k)}(\tau), \mathbf{u}^{(k)}(\tau), t) = \mathbf{0}, \quad (k = 1,\dots,K) \quad (20)$$

the path constraints

$$\mathbf{c}_{\min} \le \mathbf{c}(\mathbf{y}^{(k)}(\tau), \mathbf{u}^{(k)}(\tau), t) \le \mathbf{c}_{\max}, \quad (k = 1,\dots,K) \quad (21)$$

and the boundary conditions

$$\mathbf{b}_{\min} \le \mathbf{b}(\mathbf{y}^{(1)}(-1), t_0, \mathbf{y}^{(K)}(+1), t_f) \le \mathbf{b}_{\max}. \quad (22)$$

Because the state must be continuous at each interior mesh point, it is required that the condition $\mathbf{y}(T_k^-) = \mathbf{y}(T_k^+),\ (k = 1,\dots,K-1)$, be satisfied at the interior mesh points $(T_1,\dots,T_{K-1})$.

IV. Legendre–Gauss–Radau Collocation

The multiple-interval form of the continuous-time Bolza optimal control problem in Section III is discretized using collocation at LGR points as described in [5], [17], and [20]. In the LGR collocation method, the state of the continuous-time Bolza optimal control problem is approximated in $\mathcal{S}_k$, $k \in [1, \ldots, K]$, as

$$\mathbf{y}^{(k)}(\tau) \approx \mathbf{Y}^{(k)}(\tau) = \sum_{j=1}^{N_k+1} \mathbf{Y}_j^{(k)} \ell_j^{(k)}(\tau)$$

$$\ell_j^{(k)}(\tau) = \prod_{\substack{l=1 \\ l \neq j}}^{N_k+1} \frac{\tau - \tau_l^{(k)}}{\tau_j^{(k)} - \tau_l^{(k)}} \tag{23}$$

where $\tau \in [-1, +1]$, $\ell_j^{(k)}(\tau)$, $(j = 1, \ldots, N_k + 1)$, is a basis of Lagrange polynomials, $(\tau_1^{(k)}, \ldots, \tau_{N_k}^{(k)})$ are the LGR [5] collocation points in $\mathcal{S}_k = [T_{k-1}, T_k)$, and $\tau_{N_k+1}^{(k)} = T_k$ is a noncollocated point. Differentiating $\mathbf{Y}^{(k)}(\tau)$ in (23) with respect to τ gives

$$\frac{d\mathbf{Y}^{(k)}(\tau)}{d\tau} = \sum_{j=1}^{N_k+1} \mathbf{Y}_j^{(k)} \frac{d\ell_j^{(k)}(\tau)}{d\tau}. \tag{24}$$

Defining $t_i^{(k)} = t(\tau_i^{(k)}, t_0, t_f)$ using (18), the dynamics are then approximated at the N_k LGR points in mesh interval $k \in [1, \ldots, K]$ as

$$\sum_{j=1}^{N_k+1} D_{ij}^{(k)} \mathbf{Y}_j^{(k)} - \frac{t_f - t_0}{2} \mathbf{a}\big(\mathbf{Y}_i^{(k)}, \mathbf{U}_i^{(k)}, t_i^{(k)}\big) = \mathbf{0},$$
$$(i = 1, \ldots, N_k) \tag{25}$$

where $D_{ij}^{(k)} = d\ell_j^{(k)}(\tau_i^{(k)})/d\tau$, $(i = 1, \ldots, N_k)$, $(j = 1, \ldots, N_k + 1)$, are the elements of the $N_k \times (N_k + 1)$ *LGR differentiation matrix* [5] in mesh interval $\mathcal{S}_k$, $k \in [1, \ldots, K]$. The LGR discretization then leads to the following NLP. Minimize

$$\mathcal{J} \approx \mathcal{M}\big(\mathbf{Y}_1^{(1)}, t_0, \mathbf{Y}_{N_K+1}^{(K)}, t_f\big) + \sum_{k=1}^{K} \sum_{j=1}^{N_k} \frac{t_f - t_0}{2} w_j^{(k)} \mathcal{L}\big(\mathbf{Y}_j^{(k)}, \mathbf{U}_j^{(k)}, t_j^{(k)}\big) \tag{26}$$

subject to the collocation constraints of (25) and the constraints

$$\mathbf{c}_{\min} \leq \mathbf{c}\big(\mathbf{Y}_i^{(k)}, \mathbf{U}_i^{(k)}, t_i^{(k)}\big) \leq \mathbf{c}_{\max}, \quad (i = 1, \ldots, N_k) \tag{27}$$

$$\mathbf{b}_{\min} \leq \mathbf{b}\big(\mathbf{Y}_1^{(1)}, t_0, \mathbf{Y}_{N_K+1}^{(K)}, t_f\big) \leq \mathbf{b}_{\max} \tag{28}$$

$$\mathbf{Y}_{N_k+1}^{(k)} = \mathbf{Y}_1^{(k+1)}, \quad (k = 1, \ldots, K - 1) \tag{29}$$

where $N = \sum_{k=1}^{K} N_k$ is the total number of LGR points and (29) is the continuity condition on the state and is enforced at the interior mesh points $(T_1, \ldots, T_{K-1})$ by treating $\mathbf{Y}_{N_k+1}^{(k)}$ and $\mathbf{Y}_1^{(k+1)}$ as the same variable in the NLP.

Finally, it is noted that the mesh refinement method developed in this brief requires an estimate of the solution error on the current mesh. In this brief, the approach for estimating the solution relative error is taken from [17]. The relative error approximation derived in [16] is obtained by comparing two approximations to the state, one with higher accuracy. The key idea is that for a problem whose solution is smooth, an increase in the number of LGR points should yield a state that more accurately satisfies the dynamics. Hence, the difference between the solution associated with the original set of LGR points and the approximation associated with the increased number of LGR points should yield an approximation of the error in the state. The details of this relative error estimate are beyond the scope of this brief and the reader is referred to [17] for the details.

V. *hp*-Adaptive Mesh Refinement Method

After solving the NLP arising from the aforementioned LGR collocation method on mesh M, the maximum relative error estimate is computed in each mesh interval using the method mentioned at the end of Section IV and as described in [17]. If the maximum relative error in any mesh interval exceeds a user-specified accuracy tolerance, ϵ, then the mesh interval is modified either by increasing the degree of the approximating polynomial in the mesh interval or by dividing the mesh interval into smaller intervals. In this brief, the criteria for modifying the mesh is based on the approach described in Section II. In particular, if the coefficients of a Legendre polynomial approximation of the state decay at a rate faster than a user-defined decay rate, then the polynomial degree is increased, as described in Section V-A. On the other hand, if the Legendre polynomial coefficients decay at a rate slower than the user-defined decay rate, then the mesh interval is divided into subintervals as described in Section V-B. In addition to increasing the size of the mesh, using the procedure in [18], the mesh size can be reduced if the maximum relative error is less than the mesh refinement accuracy tolerance in one or more adjacent mesh intervals. A brief description of the mesh size reduction procedure is given in Section V-C.

A. Method for Increasing Polynomial Degree

Suppose now that the error tolerance in a given mesh interval $\mathcal{S}_k$ has not been met and that the decay rate of Legendre coefficients is greater than $\bar{\sigma}$. In this case, the solution in the mesh interval is regarded as smooth in $\mathcal{S}_k$ and, if possible, the degree of the polynomial approximation used on mesh $M + 1$ is increased in order to reduce the solution error. Let $e_k^{(M)}$ denote the error on interval $\mathcal{S}_k$ of mesh M. Treating the upper bound of the error estimate in (10) as $e_k^{(M)}$ gives the relationship

$$e_k^{(M)} = \frac{c}{\sqrt{1 - 10^{-2\sigma}}} 10^{-\sigma\left(N_k^{(M)}+1\right)}. \tag{30}$$

Furthermore, assume for the ensuing mesh $M + 1$ that it is desired to achieve a maximum relative actual error accuracy ϵ. Again, using the upper bound of the error estimate in (10) as the actual error estimate gives

$$\epsilon = \frac{c}{\sqrt{1 - 10^{-2\sigma}}} 10^{-\sigma\left(N_k^{(M+1)}+1\right)}. \tag{31}$$

Equations (30) and (31) can then be solved for $N_k^{(M+1)}$ to give

$$N_k^{(M+1)} = N_k^{(M)} + \frac{\log_{10}\left(\frac{e_k^{(M)}}{\epsilon}\right)}{\sigma} \tag{32}$$

where the value of σ is computed using the method in Section II. Now, in order to obtain a strict increase in the number of collocation points in $\mathcal{S}_k$ on mesh $M+1$, the result of (32) is replaced with

$$N_k^{(M+1)} = \left\lceil N_k^{(M)} + \frac{\log_{10}\left(\frac{e_k^{(M)}}{\epsilon}\right)}{\sigma} \right\rceil \tag{33}$$

where $\lceil \cdot \rceil$ replaces the argument with the next highest integer.

B. Method for Dividing a Mesh Interval

Assume now that the decay rate of Legendre coefficients is less than $\bar{\sigma}$ on the current mesh and that the mesh interval needs to be divided. If the decay rate of the Legendre coefficients σ is close to zero, which indicates that the error converges very slow, the small σ will result in a large mesh size on mesh $M+1$. To avoid creating an unnecessarily large mesh, the threshold $\bar{\sigma}$ is used to determine the number of subintervals on the next mesh. The sum of the number of collocation points in the newly created mesh intervals should equal the predicted polynomial degree for the next mesh in (32) using $\bar{\sigma}$. The sum of the number of collocation points in the newly created mesh intervals is obtained as

$$\tilde{N}_k = N_k^{(M)} + \frac{\log_{10}\left(\frac{e_k^{(M)}}{\epsilon}\right)}{\bar{\sigma}}. \tag{34}$$

Each newly created subinterval on mesh $M+1$ contains the same number collocation points on mesh M. Using this strategy, the number of subintervals, H_k, into which $\mathcal{S}_k$ is divided is computed as

$$H_k = \left\lceil \frac{\tilde{N}_k}{N_k^{(M)}} \right\rceil. \tag{35}$$

C. Mesh Size Reduction

In addition to the two strategies described in Sections V-A and V-B for increasing the size of the mesh, the mesh refinement method in this brief also allows for reducing the size of the mesh using the method in [18]. As described in [18], the mesh size can be reduced by either reducing the number of collocation points in a mesh interval or by reducing the number of mesh intervals. Refer to [18] for complete details of the mesh reduction method employed in this brief.

VI. *hp*-ADAPTIVE ALGORITHM

The *hp*-adaptive mesh refinement method that arises from Section V is summarized in Fig. 4, where M denotes the mesh refinement iteration. The algorithm terminates in Step 4 when the error tolerance is satisfied or when a prescribed maximum number of mesh refinement iterations, $M_{\max}$, is attained.

Step 1: Supply initial mesh.
Step 2: Solve the LGR collocation NLP on mesh M.
Step 3: Compute scaled error in each mesh interval using the the error estimate given in Ref. 17.
Step 4: If error tolerance is satisfied in all mesh intervals, then quit. Otherwise, proceed to **Step 5**.
Step 5: For every mesh interval on current mesh,
a) if error tolerance is not satisfied, modify the mesh interval using the method of Section V-A and V-B;
b) if error tolerance is satisfied, determine if the mesh size can be reduced using the method of Section V-C.
Step 6: Return to **Step 2**.

Fig. 4. *hp*-Legendre mesh refinement method.

VII. EXAMPLES

The *hp*-Legendre mesh refinement method described in Section V is now applied to two examples and the performance of this method is compared with the previously developed *ph* and *hp* methods described in [17] and [18], respectively. The following terminology is used in the forthcoming analysis. First, M denotes the number of mesh refinement iterations ($M = 0$ corresponds to the initial mesh), while N and K denote the total number of LGR collocation points and the number of mesh intervals, respectively. All results were obtained with the MATLAB optimal control software $\mathbb{GPOPS}-\mathbb{II}$ [21] using the NLP solver SNOPT [2] and an optimality tolerance of 10^{-7} with NLP solver derivatives computed using the MATLAB algorithmic differentiation tool *AdiGator* [22]. No upper limit is imposed on the allowable polynomial degree in a mesh interval, but the number of mesh refinements is limited to 25. In each example, the initial mesh consists of ten uniformly spaced mesh intervals and four collocation points per mesh interval, and the initial guess for all examples is a straight line for variables with boundary conditions at both endpoints and is a constant for variables with boundary conditions at only one endpoint. All computations were performed on a 2.3-GHz Intel Core i7 MacBook Pro running MAC OS-X version 10.10.4 (Yosemite) with 16-GB 1600-MHz DDR3 of RAM and MATLAB Version R2014b (build 8.4.0.150421) and the CPU times reported are ten-run averages (excluding the time required to solve the NLP on the first mesh).

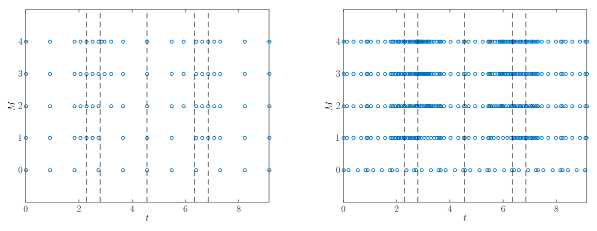
Example 1 (Minimum-Time Reorientation of a Robot Arm): Consider the following minimum-time reorientation of a robot arm taken from [23]. The objective is to move an arm from an initial position with zero velocity to a final position with zero velocity while minimizing the time taken to perform the maneuver. Mathematically, the problem is stated as follows. Minimize $J = t_f$ subject to

$$\begin{aligned} &\ddot{y}_1 = u_1/L, \quad \ddot{y}_2 = u_2/I_\theta, \quad \ddot{y}_3 = u_3/I_\phi \\ &|u_i| \leq , \quad (i = 1, 2, 3) \end{aligned} \tag{36}$$

and the boundary conditions

$$\begin{aligned} &(y_1(0), y_1(t_f)) = (9/2, 9/2), \quad (\dot{y}_1(0), \dot{y}_1(t_f)) = (0, 0) \\ &(y_2(0), y_2(t_f)) = (0, 2\pi/3), \quad (\dot{y}_2(0), \dot{y}_2(t_f)) = (0, 0) \\ &(y_3(0), y_3(t_f)) = (\pi/4, \pi/4), \quad (\dot{y}_3(0), \dot{y}_3(t_f)) = (0, 0) \end{aligned} \tag{37}$$

where $L = 5$, $I_\phi = ((L - y_1)^3 + y_1^3)/3$, and $I_\theta = I_\phi \sin^2(y_5)$. It is known for this problem that the control components u_1, u_2, and u_3 take the value $+1$ on the intervals $[2.286, 6.855]$, $[0, 4.570]$, and $[2.827, 6.385]$, respectively, and take the value -1 otherwise [that is, the control is discontinuous at



(a) (b)

Fig. 5. Mesh refinement history for Example 1 using the hp-Legendre method ($\bar{\sigma} = 0.25$) with an accuracy tolerance $\epsilon = 10^{-7}$. (a) hp-Legendre ($\bar{\sigma} = 0.25$) mesh point history. (b) hp-Legendre ($\bar{\sigma} = 0.25$) collocation point history.

$t = (2.286,\ 2.827,\ 4.570,\ 6.385, 6.855)$]. Fig. 5(a) shows the evolution of the mesh points for this example using the hp-Legendre method with $\bar{\sigma} = 0.25$. It is seen that on the first mesh iteration ($M = 1$), a few mesh points are added to the intervals that contain the discontinuities, and the first, third, and fifth discontinuities are accurately located. From the second mesh iteration ($M = 2$) to the final mesh iteration, more mesh intervals are added in the neighborhoods of the second and fourth discontinuities and no mesh interval is added to the segments $[0, 2]$ and $[7, t_f]$, where the solution is smooth. The manner in which the mesh evolves demonstrates that the decay rate of the Legendre polynomial coefficients (used as the basis of the method) is effective in identifying the regions of smoothness and regions where discontinuities in the solution exist. For this example, the decay rate of the Legendre polynomial coefficients, σ, is less than the threshold $\bar{\sigma} = 0.25$ in the neighborhoods of the discontinuities and is greater than the threshold $\bar{\sigma} = 0.25$ in the segments $[0, 2]$ and $[7, t_f]$, where the solution is smooth. Because the decay rate of the Legendre polynomial coefficients is below the threshold near the discontinuities, in these segments, the solution error is reduced by creating new mesh intervals. Contrariwise, because the decay rate of the Legendre coefficients is larger than the threshold in the segments $[0, 2]$ and $[7, t_f]$, the error in these segments is reduced by increasing the degree of the polynomial approximation. Thus, the hp-Legendre appropriately increases the degree of the approximating polynomial or refines the mesh in the regions where the solution is smooth and nonsmooth, respectively. Corresponding to the estimated decay rate, the mesh points are placed more sparsely in segments where the solution is smooth and are placed more densely in regions where the solution is nonsmooth.

Next, the computational efficiency and mesh size of the hp-Legendre method are compared with the computational efficiency and mesh size of the ph method in [17] and the hp method in [18]. Table I shows the performance of the various hp-Legendre methods alongside the ph method and the hp method for $\epsilon = (10^{-7}, 10^{-8}, 10^{-9})$. First, it is seen that using the ph method, the number of mesh iterations required to meet the mesh refinement accuracy tolerance $\epsilon = 10^{-7}$ is much greater (where $M = 24$) than that using the hp-Legendre method [where $M = (4, 5, 6, 6)$ for $\bar{\sigma} = (0.25, 0.5, 0.75, 1)$, respectively]. In the ph method, the error is decreased most often by increasing the polynomial degree as opposed to creating new mesh intervals. In this example, however, the error is the largest near the discontinuities in the control. As a result,

TABLE I

PERFORMANCE OF THE hp-LEGENDRE, ph, AND hp METHODS FOR EXAMPLE 1

$\epsilon = 10^{-7}$

Method	$\bar{\sigma}$	CPU Time (s)	N	K	M
ph	-	1.63	158	10	24
hp	-	0.47	178	24	5
hp-Legendre	0.25	0.19	116	20	4
hp-Legendre	0.5	0.33	139	26	5
hp-Legendre	0.75	0.28	100	23	6
hp-Legendre	1	0.28	101	23	6

$\epsilon = 10^{-8}$

Method	$\bar{\sigma}$	CPU Time (s)	N	K	M
ph	-	-	-	-	-
hp	-	0.82	271	25	5
hp-Legendre	0.25	0.74	249	31	7
hp-Legendre	0.5	0.46	172	33	6
hp-Legendre	0.75	0.37	163	29	6
hp-Legendre	1	0.39	138	25	6

$\epsilon = 10^{-9}$

Method	$\bar{\sigma}$	CPU Time (s)	N	K	M
ph	-	-	-	-	-
hp	-	1.64	371	46	7
hp-Legendre	0.25	0.63	335	30	6
hp-Legendre	0.5	0.48	190	29	6
hp-Legendre	0.75	0.82	236	48	8
hp-Legendre	1	0.54	201	37	7

increasing the polynomial degree in these segments results in a slow decrease in the error. Thus, the CPU time required to solve the problem using the hp-Legendre methods is less than using the ph method. Moreover, as the tolerance gets tighter [$\epsilon = (10^{-8}, 10^{-9})$], the ph method fails to solve the problem. Next, it is seen from Table I that for $\epsilon = 10^{-8}$, the number of mesh iterations required to meet the mesh refinement accuracy tolerance using the hp method (where $M = 5$) is the least among all the methods, but the CPU time is the largest. The CPU time is the largest using the hp method, because the number of collocation points in a single mesh interval obtained from the hp method is extremely large (reaching values as large as 120), because the upper bound on the LGR convergence rate (used as the basis of the hp method) results in an extremely large estimate for the required polynomial degree. On the other hand, the growth in the polynomial degree using the hp-Legendre method is much slower than that of the hp method, because the new value of the polynomial degree computed from (33) will be significantly smaller than the value computed by the hp method. Furthermore, it is seen for this example that in all the cases, the final mesh size obtained by the hp-Legendre method is smaller than the final mesh size obtained by the hp method, while the number of mesh iterations required to meet the mesh refinement accuracy tolerance, ϵ, is approximately the same for either method. Interestingly, it is also seen that the hp-method is more computationally efficient than the hp method as ϵ decreases. Thus, the CPU time using the hp-Legendre method grows more slowly as the demand for accuracy increases when compared with the hp method.

Example 2 (Minimum-Time Supersonic Aircraft Climb): Consider the following problem of minimizing the time required to transfer a supersonic aircraft from takeoff to a terminal altitude and speed. This optimal control problem, taken from [24], which uses the model in [25], is stated as

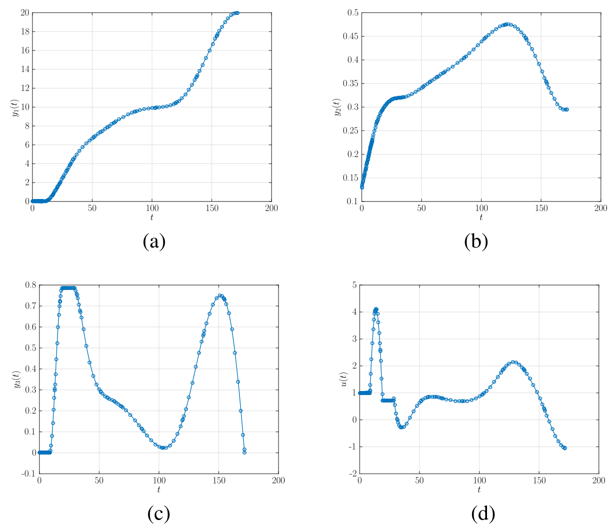


Fig. 6. Solution of Example 2 using the *hp*-Legendre method ($\bar{\sigma} = 0.5$) with an accuracy tolerance $\epsilon = 10^{-7}$. (a) $y_1(t)$ versus t. (b) $y_2(t)$ versus t. (c) $y_3(t)$ versus t. (d) $u(t)$ versus t.

follows. Minimize $J = t_f$ subject to

$$\dot{h} = v \sin\gamma, \quad \dot{v} = \frac{T - D}{m} - g\sin\gamma, \quad \dot{\gamma} = g\frac{u - \cos\gamma}{v} \tag{38}$$

the boundary conditions

$$\begin{aligned}(h(0), h(t_f)) &= (0, 20)\ \text{km}\\ (v(0), v(t_f)) &= (0.1293, 0.2951)\ \text{km}\cdot\text{s}^{-1}\\ (\gamma(0), \gamma(t_f)) &= (0, 0)\ \text{rad}\end{aligned} \tag{39}$$

and the state inequality path constraints

$$h(t) \geq 0, \quad < \gamma(t) \leq \pi/4 \tag{40}$$

where h is the altitude, v is the speed, γ is the flight path angle, $T = T(h, M)$ is the thrust force, $D = D(h, M)$ is the drag force, u is the load factor (which is the vertical component of the lift), g is the acceleration due to gravity, $M = v/a$ is the Mach number, and $a = a(h)$ is the speed of sound. This example is a challenging real-world application where the thrust and the drag are obtained from two-dimensional polynomial fits of tabular data and the speed of sound is obtained from a one-dimensional polynomial fit of tabular data. It is noted that expressions for T, D, and a are obtained from the data given in [25], and the solution to this problem is shown in Fig. 6. It is shown in Fig. 6 that the solution is least smooth in the segments where the one or both of the state inequality constraints are active. As a result, one would expect the *hp*-Legendre method to place mesh points near the switches in the activity of path constraints. Examining Fig. 7(a), where the *hp*-Legendre method is implemented with $\epsilon = 10^{-7}$ and $\bar{\sigma} = 0.5$, it is seen that the *hp*-Legendre method progressively increases the number of mesh points in the region near the path constraint activity. First, it is seen that on the first mesh iteration ($M = 1$), mesh points are added to the segment $t \in [0, 40]$, where the two aforementioned path constraints are active. As the mesh refinement proceeds, more mesh points are added to the segment where the path constraints are active and the decay rate of the Legendre polynomial coefficients drops

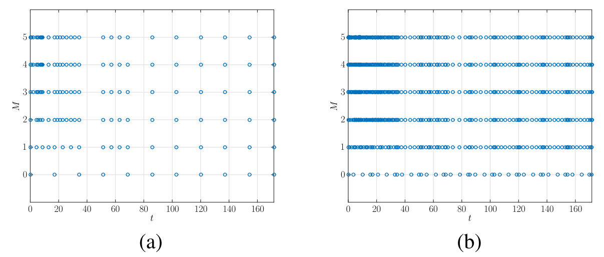


Fig. 7. Mesh refinement history for Example 2 using the *hp*-Legendre method ($\bar{\sigma} = 0.5$) with an accuracy tolerance $\epsilon = 10^{-7}$. (a) *hp*-Legendre ($\bar{\sigma} = 0.5$) mesh point history. (b) *hp*-Legendre ($\bar{\sigma} = 0.5$) collocation point history.

below $\bar{\sigma} = 0.5$, because the solution in the resulting mesh intervals becomes nonsmooth. Now, although at the start of the mesh refinement a few mesh points are added in the segment $t \in [50, 70]$ (where the solution is smooth), from the second mesh iteration ($M = 2$) onward, no more mesh points are placed in this region. Second, it is seen that the mesh points are placed sparsely in the segment $t \in [70, 110]$ (again, where the solution is smooth), and new mesh intervals are not created in this segment. As a result, in the segment $t \in [70, 110]$, the error in the solution is decreased only by increasing the polynomial degree in the existing mesh intervals.

The computational efficiency and mesh size of *hp*-Legendre mesh refinement method is now compared against the previously developed *ph* method in [17] and the *hp* method of [18], and Table II summarizes the results using mesh refinement accuracy tolerances $\epsilon = (10^{-7}, 10^{-8}, 10^{-9})$. First, it is noted that the *ph* method fails to solve this example with all the values of ϵ. Second, it is seen from Table II that for $\epsilon = 10^{-7}$, the *hp* method requires eight mesh refinement iterations to achieve the required accuracy, while the *hp*-Legendre method requires less than eight mesh refinement iterations for all values of $\bar{\sigma}$. Furthermore, $\epsilon = (10^{-8}, 10^{-9})$, and the *hp* method fails because the maximum polynomial degree obtained from the *hp* method is 814 in at least mesh interval, while the maximum polynomial degree for the *hp*-Legendre method is only 22. Thus, while the *hp* is an improvement over the *ph* method, the *hp*-Legendre method performs significantly better than the *hp* method.

VIII. Discussion

The examples illustrate different features of the *hp*-Legendre mesh refinement method. The first example shows that the method accurately locates discontinuities due to the prediction of nonsmoothness in the solution from the slow decay rate of the Legendre polynomial coefficients. The second example shows that the *hp*-Legendre mesh refinement strategy can detect nonsmoothness of the problem that includes active state path constraints. In addition, in the examples, the largest polynomial degree attained using the *hp*-Legendre method was 25, which is quite reasonable and made it unnecessary to place an upper limit on the polynomial degree. In fact, the increase in the polynomial degree, defined by the term $\log_{10}(e/\epsilon)/\sigma$ in (33), will never be very large. On the other hand, because the polynomial degrees obtained by the *ph* and *hp* methods were larger polynomial than those of the *hp*-Legendre method, the *ph* method was unable to

TABLE II

PERFORMANCE OF THE hp-LEGENDRE, ph, AND hp METHODS FOR EXAMPLE 2

$\epsilon = 10^{-7}$

Method	$\bar{\sigma}$	CPU Time (s)	N	K	M
ph	-	-	-	-	-
hp	-	2.25	203	26	8
hp-Legendre	0.25	1.93	173	24	5
hp-Legendre	0.50	1.07	162	30	5
hp-Legendre	0.75	1.48	201	43	6
hp-Legendre	1	1.42	214	51	5

$\epsilon = 10^{-8}$

Method	$\bar{\sigma}$	CPU Time (s)	N	K	M
ph	-	-	-	-	-
hp	-	-	-	-	-
hp-Legendre	0.25	5.54	345	49	12
hp-Legendre	0.50	3.21	354	56	10
hp-Legendre	0.75	2.18	334	71	7
hp-Legendre	1	6.82	465	102	18

$\epsilon = 10^{-9}$

Method	$\bar{\sigma}$	CPU Time (s)	N	K	M
ph	-	-	-	-	-
hp	-	-	-	-	-
hp-Legendre	0.25	11.01	696	93	10
hp-Legendre	0.50	12.97	774	116	14
hp-Legendre	0.75	7.78	760	127	12
hp-Legendre	1	7.75	895	174	10

attain the required mesh refinement accuracy tolerance or did so while requiring a polynomial degree as large as 183. Next, while the number of mesh intervals created by the hp and hp-Legendre methods was similar when nonsmoothness was detected, the highest polynomial degree attained by the hp method was an extremely computationally ineffcient value of 814. Finally, it is noted that the hp-Legendre method is much simpler to implement than the hp method, because it does not require initialization by comparing the solution on two meshes and is based on a simple computation of a decay rate of Legendre polynomial coefficients.

IX. CONCLUSION

An adaptive mesh refinement method for solving optimal control problems has been described. The method modifies the mesh either by increasing the polynomial degree within a mesh interval or by dividing a mesh interval into subintervals. The decision to increase the degree of the polynomial within a mesh interval or to create new mesh intervals is based on the decay rate of the coefficients of a Legendre polynomial approximation of the state as a function of the index of the Legendre polynomial expansion, where it has been shown that the decay rate of these coefficients is the same as the decay rate on the upper bound of the state approximation error. In addition, the method allows for mesh size reduction using a previously developed mesh reduction technique. The foundation of the method has been provided, the key components of the method have been described, and the method has been demonstrated on two examples. It is found that the approach developed in this brief is more efficient, simpler to implement, and requires the specification of fewer user-defined parameters when compared with recently developed adaptive mesh refinement methods for optimal control.

REFERENCES

[1] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, 2nd ed. Philadelphia, PA, USA: SIAM, 2009.

[2] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM Rev.*, vol. 47, no. 1, pp. 99–131, Jan. 2002.

[3] G. Elnagar, M. A. Kazemi, and M. Razzaghi, "The pseudospectral Legendre method for discretizing optimal control problems," *IEEE Trans. Autom. Control*, vol. 40, no. 10, pp. 1793–1796, Oct. 1995.

[4] D. A. Benson, G. T. Huntington, T. P. Thorvaldsen, and A. V. Rao, "Direct trajectory optimization and costate estimation via an orthogonal collocation method," *J. Guid. Control Dyn.*, vol. 29, no. 6, pp. 1435–1440, Dec. 2006.

[5] D. Garg *et al.*, "Direct trajectory optimization and costate estimation of finite-horizon and infinite-horizon optimal control problems via a radau pseudospectral method," *Comput. Optim. Appl.*, vol. 49, no. 2, pp. 335–358, Jun. 2011, doi: 10.1007/s10589–00–09291–0.

[6] W. W. Hager, H. Hou, and A. V. Rao, "Lebesgue constants arising in a class of collocation methods," *IMA J. Numer. Anal.*, 2016, doi: 10.1093/imanum/drw060.

[7] W. W. Hager, H. Hou, and A. V. Rao, "Convergence rate for a Gauss collocation method applied to unconstrained optimal control," *J. Optim. Theory Appl.*, vol. 169, no. 3, pp. 801–824, 2016.

[8] W. W. Hager, H. Hou, and A. V. Rao. (2015). "Convergence rate for a radau collocation method applied to unconstrained optimal control." [Online]. Available: https://arxiv.org/abs/1508.03783

[9] W. W. Hager, H. Hou, S. Mohapatra, and A. V. Rao. (2016). "Convergence rate for an hp collocation method applied to unconstrained optimal control." [Online]. Available: https://arxiv.org/abs/1605.02121

[10] W. W. Hager, S. Mohapatra, and A. V. Rao. (2016). "Convergence rate for a Gauss collocation method applied to constrained optimal control." [Online]. Available: https://arxiv.org/abs/1607.02798

[11] Q. Gong, F. Fahroo, and I. M. Ross, "Spectral algorithm for pseudospectral methods in optimal control," *J. Guid., Control, Dyn.*, vol. 31, no. 3, pp. 460–471, May 2008.

[12] Y. Zhao and P. Tsiotras, "Density functions for mesh refinement in numerical optimal control," *J. Guid., Control, Dyn.*, vol. 34, no. 1, pp. 271–277, Feb. 2011.

[13] I. Babuska and M. Suri, "The p and hp version of the finite element method, an overview," *Comput. Methods Appl. Mech. Eng.*, vol. 80, pp. 5–26, Mar. 1990.

[14] H. Wang and S. Xiang, "On the convergence rates of legendre approximation," *Math. Comput.*, vol. 81, no. 278, pp. 861–877, 2012.

[15] C. Mavriplis, "Adaptive mesh strategies for the spectral element method," *Comput. Methods Appl. Mech. Eng.*, vol. 116, no. 1, pp. 77–86, 1994.

[16] W. F. Mitchell and M. A. McClain, "A comparison of hp-adaptive strategies for elliptic partial differential equations," *ACM Trans. Math. Softw.*, vol. 41, no. 1, pp. 2:1–2:39, Oct. 2014. [Online]. Available: http://doi.acm.org/10.1145/2629459

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

[18] F. Liu, W. W. Hager, and A. V. Rao, "Adaptive mesh refinement for optimal control using nonsmoothness detection and mesh size reduction," *J. Franklin Inst.*, vol. 352, no. 10, pp. 4081–4106, Oct. 2015.

[19] J. Hesthaven and R. Kirby, "Filtering in legendre spectral methods," *Math. Comput.*, vol. 77, no. 263, pp. 1425–1452, 2008.

[20] D. Garg *et al.*, "A unified framework for the numerical solution of optimal control problems using pseudospectral methods," *Automatica*, vol. 46, no. 11, pp. 1843–1851, 2010.

[21] M. A. Patterson and A. V. Rao, "$\mathbb{GPOPS}-\mathbb{II}$, a MATLAB software for solving multiple-phase optimal control problems using hp-adaptive Gaussian quadrature collocation methods and sparse nonlinear programming," *ACM Trans. Math. Softw.*, vol. 41, no. 1, pp. 1:1–1:37, Oct. 2014.

[22] M. J. Weinstein and A. V. Rao, "A source transformation via operator overloading method for the automatic differentiation of mathematical functions in MATLAB," *ACM Trans. Math. Softw.*, vol. 42, no. 1, p. 11, Jun. 2015.

[23] E. Dolan, J. J. More, and T. S. Munson, "Benchmarking optimization software with CPOPS 3.0," Argonne Nat. Lab., Argonne, IL, USA, Tech. Rep. ANL/MCS-273, 2004.

[24] C. L. Darby, W. W. Hager, and A. V. Rao, "Direct trajectory optimization using a variable low-order adaptive pseudospectral method," *J. Spacecraft Rockets*, vol. 48, no. 3, pp. 433–445, Jun. 2011.

[25] H. Seywald, E. M. Cliff, and K. H. Well, "Range optimal trajectories for an aircraft flying in the vertical plane," *J. Guid., Control, Dyn.*, vol. 17, no. 2, pp. 389–398, Apr. 1994.