A Mesh Refinement Algorithm for Solving Optimal Control Problems Using Pseudospectral Methods

Christopher L. Darby[*]
Anil V. Rao[†]

University of Florida
Gainesville, FL 32611

A state approximation-mesh refinement algorithm is presented for determining solutions to optimal control problems using pseudospectral methods. In the method presented in this paper the polynomial approximation of the state is used to assess the difference between consecutive selections of the number of segments and the number of collocation point within each segment to determine better combinations of segments and collocation points on the subsequent grid. This process of computing the difference between polynomial approximations is continued until the difference lies below a user-specified threshold. Because the method developed in this paper combines dividing the problem into segments and determining the best number of collocation points within each segment, the approach conceived here is termed a *hybrid global/local collocation method*. The user-specified parameters of the method are provided and the approach is demonstrated successfully on four optimal control problems of varying complexity. It is found that the hybrid approach developed in this paper leads to a greater accuracy with lower overall computational time as compared to using a purely global approach.

I. Introduction

In recent years, direct collocation methods have become ubiquitous in the numerical solution of nonlinear optimal control problems. Direct collocation methods for solving optimal control problems fall into two general categories: local collocation and global collocation. In a local method, low degree polynomial approximations are used and the problem is divided into a large number of finite elements (or meshes).[1] Typically, third-degree or fourth-degree polynomials are used in local collocation and the degree of the polynomial is the same from segment to segment (leading to methods such as Hermite-Simpson and Runge-Kutta). The mesh is then refined until the local error lies within a specified tolerance.[1–3] Each mesh refinement iteration changes the location of the collocation points or add collocation points so that the finite elements are denser in regions where increased solution accuracy is needed. A second approach for using low degree polynomials is that

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

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

described in Ref. 4, where higher-order local methods (fourth-degree and fifth-degree polynomials) were found to produce more accurate solutions with equivalent number of collocation points as compared with lower-degree polynomials. In contrast, in a global method the number of meshes is held constant (typically a single mesh is used) and the number of collocation points is increased until the desired accuracy is achieved.

A particular class of collocation methods that has received increasing attention in the last decade is the class of *global orthogonal collocation methods* or *pseudospectral methods* (PS methods).[5–12] In a pseudospectral method, the state is approximated using an appropriate set of global trial (basis) functions (e.g., Lagrange or Chebyshev polynomials) and the dynamics are orthogonally collocated (i.e., the collocation points are the roots of an orthogonal polynomial). Using such a combination of global interpolating polynomials and collocation points leads to discretizations with a simple structure that converge spectrally[13] (i.e., at an exponential rate) for infinitely smooth problems. The most well developed pseudospectral methods are the *Lobatto pseudospectral method*[5] (LPM), the *Gauss pseudospectral Method* (GPM),[7,8] and the *Radau pseudospectral method*[11,12] (RPM), where collocation is performed at the Legendre-Gauss-Lobatto (LGL) points, the Legendre-Gauss (LG) points, and the Legendre-Gauss-Radau (LGR) points, respectively. All three sets of collocation points (LGL, LG, and LGR) arise from a different form of Gaussian quadrature[14] and the state is approximated globally by a Lagrange polynomial. The continuous-time optimal control problem is then transcribed to a nonlinear programming problem (NLP) to be solved by any number of well known solvers.[15–17]

Because pseudospectral methods converge spectrally for infinitely smooth problems and are easy to implement using global collocation, the vast majority of work on pseudospectral methods has focused on *global collocation*.[5–8, 12] As it turns out, however, relying solely on global collocation has several limitations. First, even for many smooth problems an accurate approximation may only be obtained by using a very high-degree global polynomial. Second, many optimal control problems have either nonsmooth solutions or nonsmooth problem formulations. Such problems may have solutions that either have discontinuities in the state and/or control, or nonsmooth changes in behavior due to active and inactive path constraints. When a problem requires a large (or perhaps infinite) degree polynomial to accurately approximate the state, the convergence rate can be extremely slow using a global polynomial. In addition, when the required degree of the polynomial approximation is extremely high, a global technique may produce a poor approximation.

In addition to the fact that the solution to many optimal control problems are not accurately approximated using a global pseudospectral method, an increasingly large-degree global polynomial negatively impacts the associated nonlinear programming problem (NLP) that arises from the pseudospectral discretization. First, as the degree of the global polynomial approximation grows, the density and size of the NLP constraint Jacobian (and, possibly, the Hessian of the Lagrangian of the NLP) also grows. As a result, solving the problem becomes computationally inefficient for a sufficiently large number of globally collocated points. On the other hand, utilizing a local pseudospectral method (i.e., a pseudospectral method using low-degree polynomials over many small segments), spectral convergence is lost for problems with smooth solutions.[18]

A natural question that arises from the previous discussion is how to devise a computational approach that divides a problem into a number of segments such that the polynomial used to approximate the solution in each segment is of sufficiently large but “reasonable” degree. The first benefit of devising such a scheme is it would utilize a polynomial of high enough degree to capture the behavior in each segment. A second benefit is that the NLP will not become increasingly intractable as the number of collocation points is increased. Thus, such an approach would have computational efficiency closer to that of solving the problem using a local method, but would maintain a larger portion of the spectral accuracy associated with using a global pseudospectral method.

Many grid refinement techniques have been previously developed. These techniques work largely on the goal of obtaining a specified solution accuracy by increasing the number of collocation points in areas of the trajectory with the largest error. An excellent example of a local collocation grid refinement method is that given in Ref. 19, where errors are analyzed on a particular grid and the grid is refined based on local errors. If errors are sufficiently large in a particular region, the number of local segments in that region is increased. Another local grid refinement technique is that of Ref. 2, where local smoothness of a solution via interpolation by neighboring points is assessed. In regions where a function is not locally smooth, grid points are added. In the context of pseudospectral methods, Ref. 20 has conducted a study on refining a global pseudospectral solution by analyzing the time derivative of the state or control. First, a suitable global approximation is obtained. The total number of collocation points is then held constant and the problem is segmented at locations where the time derivatives are large. Because the collocation points in a pseudospectral method are denser near the ends of a segment, the algorithm of Ref. 20 provides a denser collection of collocation points at the points of the segment breaks.

In this paper, a new approach for segment decomposition and node allocation using pseudospectral methods is presented. The approach developed in this paper analyzes the differences between the interpolating polynomial approximations used on successive iterates. Using this approach, it is possible to determine the regions of the trajectories where the difference in the interpolating polynomials is sufficiently large; such large differences indicate that the state approximation is not of sufficiently large degree. The algorithm uses a two-tiered strategy as follows. If the error across a particular segment has a uniform behavior, the degree of the polynomial is increased. If the error at certain points is significantly larger than errors at other points in a segment, a segment is divided into sub-segments at the points where the differences are large. Because the method developed in this paper simultaneously increases the number of segments or the degree of the polynomial a segment it is a *hybrid local/global approach.* The method devised in this paper has an advantage that it is simple to implement, does not first require a suitable initial global approximation, and maintain as much as possible the key property of spectral accuracy within each segment. This method differs from previously developed approaches in that the goal is not simply to densely locate collocation points in problematic areas. Instead, the objective of this algorithm is to determine the correct degree polynomial being used in each segment. The algorithm is demonstrated on four examples of varying complexity and is found to be a viable method for efficiently and accurately solving complex optimal control problems using pseudospectral methods.

II. Optimal Control Problem in Bolza Form

Without loss of generality, consider the following optimal control problem in Bolza form. Minimize the cost functional

$$J = \mathbf{\Phi}(x(-1), t_0, \mathbf{x}(-1), t_f) + \frac{t_f - t_0}{2} \int_{-1}^{1} \mathcal{L}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau)\, d\tau \tag{1}$$

subject to the dynamic constraints

$$\frac{d\mathbf{x}}{d\tau} = \frac{t_f - t_0}{2} f(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) \tag{2}$$

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

$$\boldsymbol{\phi}(\mathbf{x}(-1), t_0, \mathbf{x}(1), t_f) = \mathbf{0} \tag{3}$$

and the inequality path constraints

$$\mathbf{C}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau; t_0, t_f) \leq \mathbf{0} \tag{4}$$

where $\mathbf{x}$ is the state, $\mathbf{u}$ are the controls and τ is time. It is noted that the time interval $\tau \in [-1, 1]$ can be transformed to the time interval $t \in [t_0, t_f]$ by the affine transformation:

$$t = \frac{t_f - t_0}{2}\tau + \frac{t_f + t_0}{2} \tag{5}$$

III. State Approximation Using Pseudospectral Methods

In a pseudospectral method, the state is approximated using a basis of global polynomials as

$$\mathbf{x}(\tau) \approx \mathbf{X}(\tau) = \sum_{i=K}^{K+M} \mathbf{c}_i \psi_i(\tau) \tag{6}$$

where $\tau \in [-1, 1]$, $\mathbf{c}_i$, $(i = K, \ldots, K+M)$ are coefficients of the approximation, and $\psi_i(\tau)$, $(i = K, \ldots, K+M)$ are a set of basis (trial) polynomials. In many pseudospectral methods used today (particularly pseudospectral methods based on Legendre-Gaussian quadrature rules) the basis functions are Lagrange polynomials. The Lagrange polynomials corresponding to Eq. (6) are defined as

$$L_j(\tau) = \prod_{\substack{i=K \\ i \neq j}}^{K+M} \frac{\tau - \tau_i}{\tau_j - \tau_i} \tag{7}$$

where τ_i, $(i = K, \ldots, K+M)$ are the support points of the polynomials. It is known that the Lagrange polynomials satisfy the property that

$$L_i(\tau_j) = \delta_{ij} \tag{8}$$

where δ_{ij} is the Kronecker Delta function.

Suppose now that for any pseudospectral scheme we define the collocation points as $\tau_1, \ldots, \tau_N$. Using this convention, in the Gauss pseudospectral method[7,8,10] we have $K = 0$ and $M = N$ (see Refs. 7–10 for details). In the Radau pseudospectral method,[11,12] we have $K = 1$ and $M = N + 1$ (see Refs. 11 and 12 for details). Finally, the Lobatto pseudospectral method[5] we have $K = 1$ and $M = N$ (see Ref. 5 for details). The general Lagrange polynomial state approximation can then be written as

$$\mathbf{x}(\tau) \approx \mathbf{X}(\tau) = \sum_{i=K}^{K+M} \mathbf{X}(\tau_i) L_i(\tau) \tag{9}$$

where $\mathbf{X}(\tau_i) = \mathbf{x}(\tau_i),\ (i = K, \ldots, K+M)$ is the value of the state at the support point. Differentiating Eq. (9), we obtain

$$\dot{\mathbf{x}}(\tau) \approx \dot{\mathbf{X}}(\tau) = \sum_{i=K}^{K+M} \mathbf{x}(\tau_i)\dot{L}_i(\tau) \tag{10}$$

The derivative of the state approximation given in Eq. (10) is then collocated at the N collocation points. Treating all vector quantities as row vectors, the collocation conditions can be written as

$$\mathbf{DX} - \frac{t_f - t_0}{2}\mathbf{F} = \mathbf{0} \tag{11}$$

where $\mathbf{D}$ is the associated differentiation matrix,[*]

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_K \\ \vdots \\ \mathbf{X}_{K+M} \end{bmatrix} \quad , \quad \mathbf{F} = \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_N \end{bmatrix}, \tag{12}$$

and $\mathbf{f}_i$ is the value of the right-hand side evaluated at the i-th collocation point.

While the aforementioned approach for approximation, differentiation, and collocation is common to most pseudospectral methods, a key assumption when using a pseudospectral method is that a solution to the NLP is a good approximation to the solution of the optimal control problem. This assumption is only valid if the approximating global polynomial is of sufficiently large degree. If, however, the approximating polynomial is of too small a degree, the right-hand side of Eq. (11), when evaluated at the values of the exact solution corresponding to the support points, will not be zero. In addition, a sufficiently large degree polynomial approximation will result not only in an accurate solution *at* the support points, but will also result in an accurate solution *in between* the support points. In other words, a sufficiently large-degree polynomial approximation can be used as an interpolant to obtain the solution at an arbitrary point in the domain. Using a global polynomial, the common practice is to increase the number of support points until the required accuracy is obtained.

When dealing with problems that are not well approximated by polynomials of "reasonable" degree, the approximation of Eq. (11) is misleading because the true solution is a polynomial of much larger degree than is used in the approximation. In the limiting case of a problem whose solution is nonsmooth, no finite-degree polynomial will suffice in approximating the true solution. Moreover, as the requirement on the degree of the polynomial increases, the NLP becomes increasingly intractable computationally.

The issue that arises from using a Lagrange polynomial can be seen by analyzing a scalar function $f(\tau)$ as follows. Suppose that $f(\tau)$ has $n+1$ continuous derivatives. Then the error between a Lagrange polynomial of degree n, $L_n(\tau)$, and the function $f(\tau)$ is given as[21]

$$E(\tau) = f(\tau) - L_n(\tau) = \frac{(\tau - \tau_0)\ldots(\tau - \tau_n)}{(n+1)!} f^{(n+1)}(\zeta_\tau) = \frac{A(\tau)}{(n+1)!} f^{(n+1)}(\zeta_\tau) \tag{13}$$

where $\zeta_\tau \in [-1, 1]$. The error in the derivative of the Lagrange polynomial and the derivative of the function at the i-th support point is then given as

$$\dot{E}(\tau_i) = \frac{\dot{A}(\tau_i)}{(n+1)!} f^{(n+1)}(\zeta_\tau) \tag{14}$$

[*] The structure of the differentiation matrices for the Lobatto, Gauss, and Radau pseudospectral methods can be found in Ref. 5 (Lobatto), Refs. 7–10 (Gauss), and Refs. 11 and 12 (Radau).

where $\dot{A}(\tau_i)$ is

$$\dot{A}(\tau_i) = (\tau_i - \tau_0)(\tau_i - \tau_1)...(\tau_i - \tau_{i-1})(\tau_i - \tau_{i+1})...(\tau_i - \tau_n) \tag{15}$$

From Eq. (14) and (15), we see that the error between the derivative of the Lagrange polynomial and the derivative of the exact function at the support points is non-zero unless $f^{n+1}(\zeta_\tau)$ is zero. Then when evaluated at the exact solution, Eq. (11) is given as

$$\mathbf{DX} - \frac{t_f - t_0}{2}\mathbf{F} = \dot{\mathbf{E}}(\boldsymbol{\tau}) \tag{16}$$

If Eq. (16) was used instead of Eq. (11) as the collocation condition, then for an approximating polynomial of any degree the solution at the collocation points would be calculated correctly. It is noted however that Eq. (16) cannot be used because $f^{n+1}(\zeta_\tau)$ is not known.

Practically speaking Eq. (11) is only satisfied to some specified tolerance. For a given value of n, two strategies can be employed to drive the right hand side to zero:

1. Reduce the magnitude of $\dot{A}(\tau_i)$ at each support point, i.e., devise a new set of support points.
2. Reduce the magnitude of $f^{n+1}(\zeta_\tau)$ at each support point, i.e., ensure that the approximating polynomial is of sufficiently high degree.

In this paper, we develop a multi-domain algorithm to obtain accurate approximations to the solution of an optimal control problem using pseudospectral methods by strategy two. The algorithm developed in this paper is designed in such a manner that Eq. (11) results in an accurate approximation to the exact solution. The basis of the algorithm is a two-tiered strategy of dividing the problem into an appropriate number of segments and determining a sufficiently large but reasonable degree polynomial in each segment. In essence, the approach devised in this paper ensures that $f^{n+1}(\zeta_\tau) \approx 0$, thus further ensuring that the error in the derivative of the Lagrange polynomial approximation is small. The approach developed in this paper leads to an an accurate approximation throughout the domain of the independent variable.

IV. Segment Decomposition and Collocation Point Allocation Algorithm

Based on the discussion given in Section III, consider again the Bolza optimal control problem as defined in Section II. Furthermore, let the trajectory be divided into S segments and let $[\tau_{s-1}, \tau_s]$ be the time interval corresponding to section $s \in [1, \ldots, S]$. Finally, let $\mathbf{X}_s(\tau)$ be the approximation to the state in segment s using N_s support points and having the form of Eq. (9). In other words, segment s is approximated by a global polynomial of degree N_s.

The objective of the algorithm is to simultaneously determine if a segment needs to be divided into more segments or if the degree of the approximating polynomial needs to be increased within a segment. The foundation of the algorithm of this paper is as follows. Consider a particular segment $s \in [1, \ldots, S]$. Furthermore, let $\mathbf{X}_s^{(k)}(\tau)$ and $\mathbf{X}_s^{(k+1)}(\tau)$ be two approximations for the state in segment s that have been determined by solving the associated NLP from the pseudospectral method. The difference between approximations using two different values of collocation points can take two different forms. In the first form, the difference is uniform across the interval. In this case no particular point in the domain accounts for the error and the algorithm is designed to increase the degree of the polynomial approximation in the segment. In the second form, the difference is significantly larger at isolated points as compared with the differences at other points within a segment. In this second case, the trajectory will be divided into segments at the points where the difference is large. The algorithm will continue to divide the problem into segments and increase the degree of polynomial within a segment until the L_∞ difference between the approximations lies below a user-specified tolerance.

To determine whether a segment has "uniform" difference or isolated points which account for the majority of the error, a user-specified ratio, ρ, is used. Let $e_s(\tau) = |\mathbf{X}_s^{(k)}(\tau)$ - $\mathbf{X}_s^{(k+1)}(\tau)|$ be the absolute difference between approximations k and $k+1$ in segment s. Next, let $\bar{e}_s$ be the mean value of $e_s(\tau)$, defined as

$$\bar{e}_s = \frac{\int_a^b \ e_s(\tau)\, d\tau}{b-a} \tag{17}$$

and let $e_s^{(i)},\ i = 1, \ldots, I$ be the number of local maxima of $e_s(\tau)$. Then, if

$$\frac{e_s^{(i)}}{\bar{e}_s} \leq \rho \tag{18}$$

for all values of $i \in [1, \ldots, I]$, segment s is said to have a "uniform" difference. On the other hand, if for any local maximum of $e_s(\tau)$ Eq. (18) is *not* satisfied, then the value of τ where Eq. (18) is violated is a place where a segment break should be placed. Using this approach, the value of ρ is a tuning parameter that weights the algorithm between being a local or a global strategy. For large values of ρ, the algorithm employ global collocation because none of the local maxima will be large enough to result in segmentation. For smaller values of ρ, the algorithm will become increasingly local because any maxima of $e_s(\tau)$ will be larger than ρ.

Assume now that a particular segment has sufficiently uniform errors between successive grids so that no further segmentation is required. The following strategy is used to determine if more collocation points are required in this segment and the amount by which the number of collocation points should be increased. Using the fact that the errors in the segment are sufficiently uniform, a "global" polynomial approximation in the segment should converge spectrally. The number of collocation points required on the next grid is then estimated via a linear extrapolation using the errors obtained on the previous grids. In particular, for example, suppose that the errors obtained on grids $k-1$ and k are $\mathcal{O}(10^{-m_{k-1}})$ and $\mathcal{O}(10^{-m_k})$, respectively, and we are interested in achieving an accuracy of $\mathcal{O}(\epsilon)$. Assuming that $m_k > m_{k-1}$, the number of collocation points estimated on the next grid would be

$$N_{k+1} = N_k + \frac{N_k - N_{k-1}}{m_k - m_{k-1}}(|\log_{10}(\epsilon)| - m_k) \tag{19}$$

The aforementioned process of increasing the number of collocation points is repeated until the desired accuracy is achieved. Eq. 19 may result in a value of N_{k+1} which is computationally too large. For problems where N_{k+1} is too large, instead of using the value of N_{k+1}, the segment should instead be segmented at the location of maximum difference. For the examples of this paper, this concern was not encountered. The algorithm below provides an adaptive method to update how many collocation points are used per segment, and also how to segment and where to locate the segments.

Algorithm: Segment Decomposition and Collocation Point Allocation

I. Initialize the problem choosing M collocation points, where M is chosen by the user.

II. *Solve problem using M and $M+1$ collocation points per segment and compare differences between the Lagrange polynomial approximations.*

III. *Determine the state with the maximum relative difference. If the maximum difference lies below the user-specified tolerance, ϵ, then terminate the algorithm. Otherwise, continue to Step IV.*

IV. *If Eq. (18) is satisfied, then increase the number of collocation points in the segment. On the first grid, increase M by 10 and return to Step II. If this is not the first time through, then compare the differences between the solutions at M and $M+10$ and extrapolate to a new number of collocation points using Eq. (19). If Eq. (19) increases the size of the segment by less than 10, then increase by 10.*

V. *If Eq. (18) is not satisfied, divide the problem into segments at all local maxima where Eq. (18) is violated, set $M = 5$ in each new segment, and return to step II.*

VI. *Terminate when the difference in all segments is below ϵ.*

Remarks:

- After a segment converges it must still be checked because changing segments around it may influence its behavior.
- We use the L_∞ difference instead of the L_1 difference because of the oscillatory nature of the Lagrange polynomials. The sum of errors between iterations may provide false positives due to cancellations.
- Changing the grid by 1 node is used to assess solution accuracy because it was found that although while introducing more iterations, this is the most robust tactic to proceed with.
- The previous solution is used as an initial guess to subsequent problems because it provides a good guess. Also, a low number of nodes is typically used to initialize because this problem is easily solved for crude guesses.
- Because of errors in approximating the locations of difficult features via low node approximations, if a segment connection is placed at t_{seg} and on a subsequent iteration, another segment is predicted at $t_{seg} \pm \delta$, the existing segment location is allowed to migrate to this new location. Multiple segments are not required here but instead are detecting the same problematic feature.

Four examples each with independent features demonstrating this method are provided. Because the predominant idea in pseudospectral methods is to use them globally, solutions obtained with this algorithm will be compared vs solutions obtained by setting ρ to be very large (i.e., global solutions).

V. Examples

We now apply the segment decomposition and node allocation algorithm of Section IV to several examples. All of the examples analyzed in this section were solved using the open-source implementation of the Gauss pseudospectral method[7–9] (GPM), GPOPS,[10,22†] using the NLP solver SNOPT.[23] It is noted that GPOPS was modified to include the algorithm of Section IV. In addition, all computations were performed using a 2 GHz Core 2 Duo machine with 2 gigabytes of RAM running OpenSuse Linux 11.0 and MATLAB R2007a.

Example 1: Hyper-Sensitive Problem

Consider the following optimal control problem. Minimize the cost functional

$$J = \frac{1}{2}\int_0^{t_f} (x^2 + u^2) dt \tag{20}$$

subject to

$$\dot{x} = -x^3 + u, \quad x(0) = x(t_f) = 1, \quad t_f = \text{Fixed} \tag{21}$$

For sufficiently large values of t_f, the solution to this example exhibits a so called “take-off”, “cruise”, and “landing” structure where the interesting behavior occurs near the initial and final time. In particular, the “cruise” segment of this trajectory is constant (i.e., state, control and, interestingly, costate) and becomes an increasingly large percentage of the total trajectory time as t_f increases.[‡] Given the structure of the solution, one would expect that the majority of collocation points would be placed in the “take-off” and “landing” segments while few collocation points would be placed in the “cruise” segment.

Suppose we use a global polynomial to approximate the solution. Fig. 1 shows the solution for $t_f = 40$ using three different approximations. It is seen that an insufficient degree polynomial (5 and 10 collocation points) results in an incorrect solution (the solution is incorrect in the ”take off” and ”landing” segments and also oscillates around the true solution in the “cruise” segment). On the other hand, utilizing a polynomial of sufficient degree (25 collocation points) results in a solution which is in excellent agreement with the optimal solution. Thus, in the case $t_f = 40$ any global polynomial of degree 25 or higher should produce an accurate solution, while a global polynomial of low degree will produce an inaccurate solution. Therefore, it would be expected that, as t_f increases, the degree of the polynomial required to solve the problem will also increase.

This example was solved using the algorithm of Section IV for different values of t_f using global collocation, $\epsilon = 10^{-3}$, and an initial grid of five collocation points. Fig. 2, 3a, and 3b display the solution obtained for $t_f = 4000$ and 290 collocation points. Many nodes are utilized in the ”cruise” segment and very few are allocated to the boundary layers. Table 1 is a summary of the number of collocation points obtained on the final grid along with computational time and number of grid refinements. It is seen in all cases that only two grid refinements were required (i.e., a total of three grids). In addition, it is noted that the prediction obtained from Eq. (19) resulted in a sufficiently large degree polynomial on the third grid that only three total grids were required. Consequently, while a very high-degree polynomial was used on the last grid, the computational time is seen to be quite reasonable. Finally, it is noted that we were unable to solve the problem using global collocation for $t_f > 4000$. For this problem, 290 nodes was not a significant computational burden because of the low dimensionality of the problem (one component of the state and one control). In

[†] GPOPS[10,22] is freely available at http://www.sourceforge.net/projects/gpops.

[‡] It is noted that this problem has been termed “hyper-sensitive”[24] due to the fact that the Hamiltonian boundary-value problem (HBVP) becomes increasingly difficult to solve using a shooting method (see Ref. 24 for details)

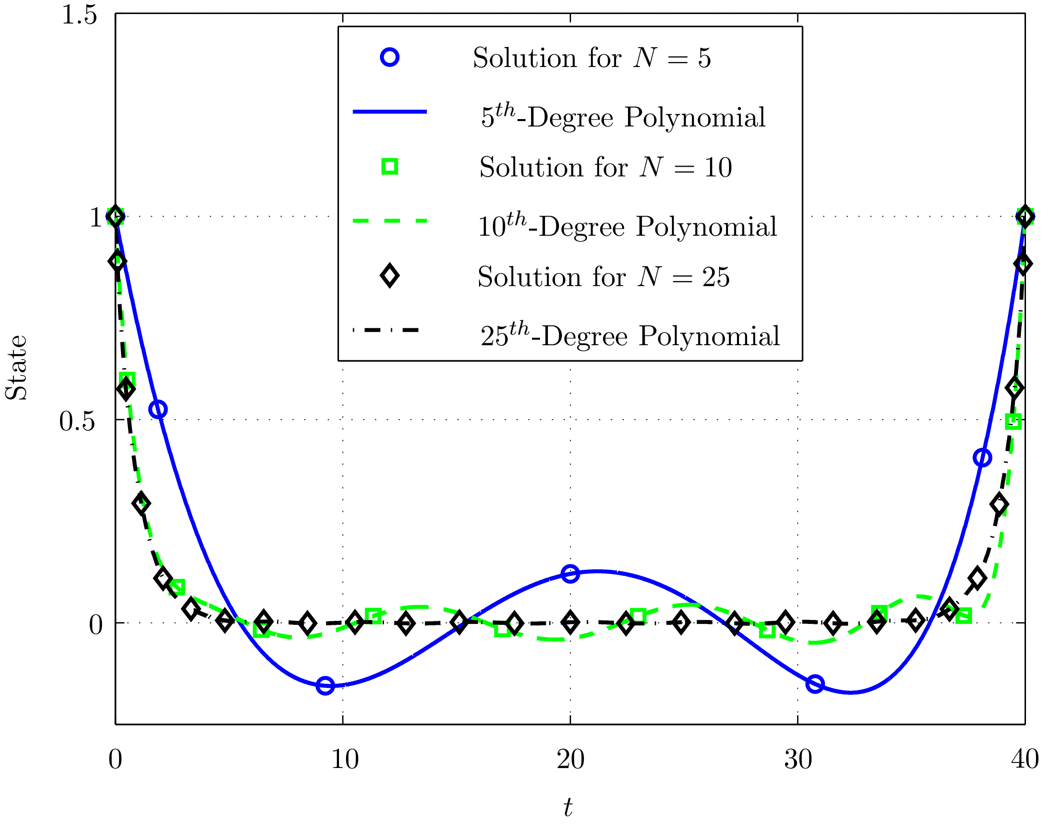


Figure 1: State vs. Time Obtained by Solving NLP for Various Values of N Alongside Underlying Lagrange Polynomial Approximation for Example with $t_f = 40$.

general, for problems with higher dimensionality, a 290 node global approximation would be too much of a computational burden and segmentation would be required.

Suppose now that the grid refinement algorithm is applied to this example for different values of t_f and $\rho = 2.5$ (thus, forcing the problem to be divided into segments). Table 2 show the performance of the algorithm. It is seen that more mesh refinements are required for large values of t_f, but significantly fewer total collocation points are used. In addition, Figs. 4, 5a, and 5b show the solution for $t_f = 5000$. As one might expect, the majority of collocation points are placed in the take-off and landing segments, while many fewer collocation points are placed in the cruise segment. With regard to computation time, it is seen that the total computation time is on par with global collocation. It is noted, however, that a solution was obtained for $t_f = 5000$ using $\rho = 2.5$ (whereas no solution was obtained using global collocation).

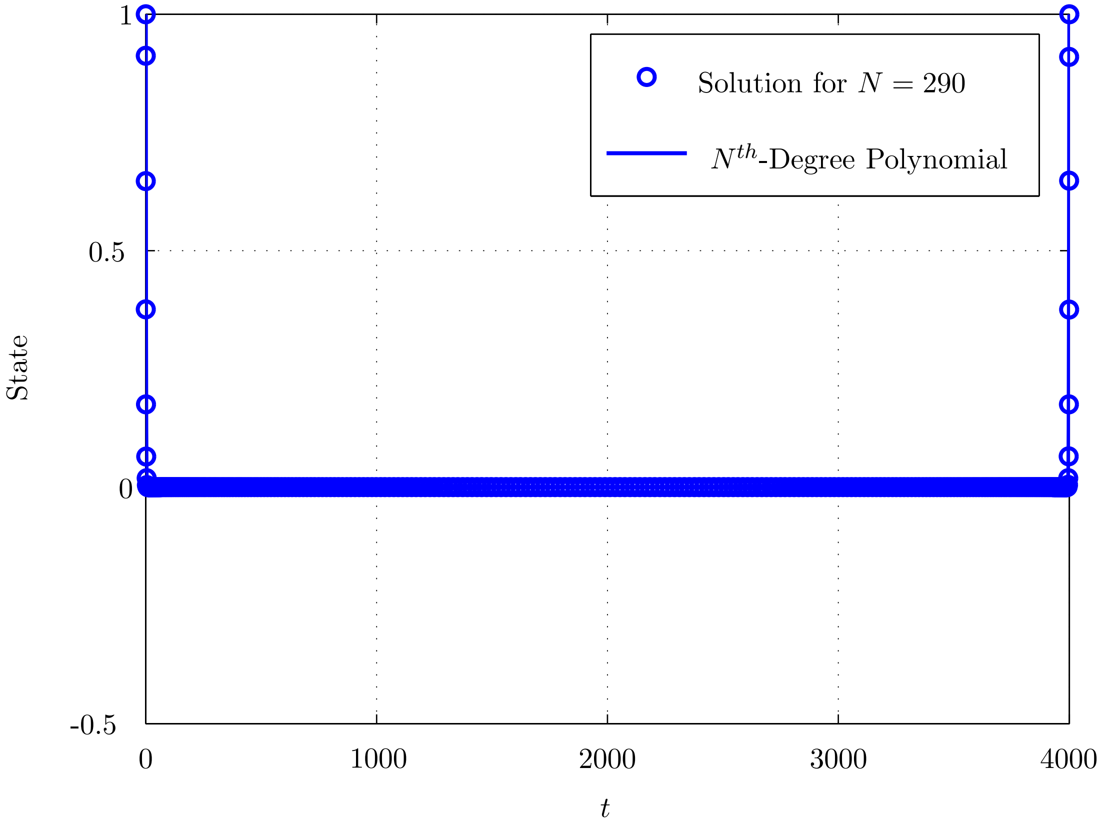
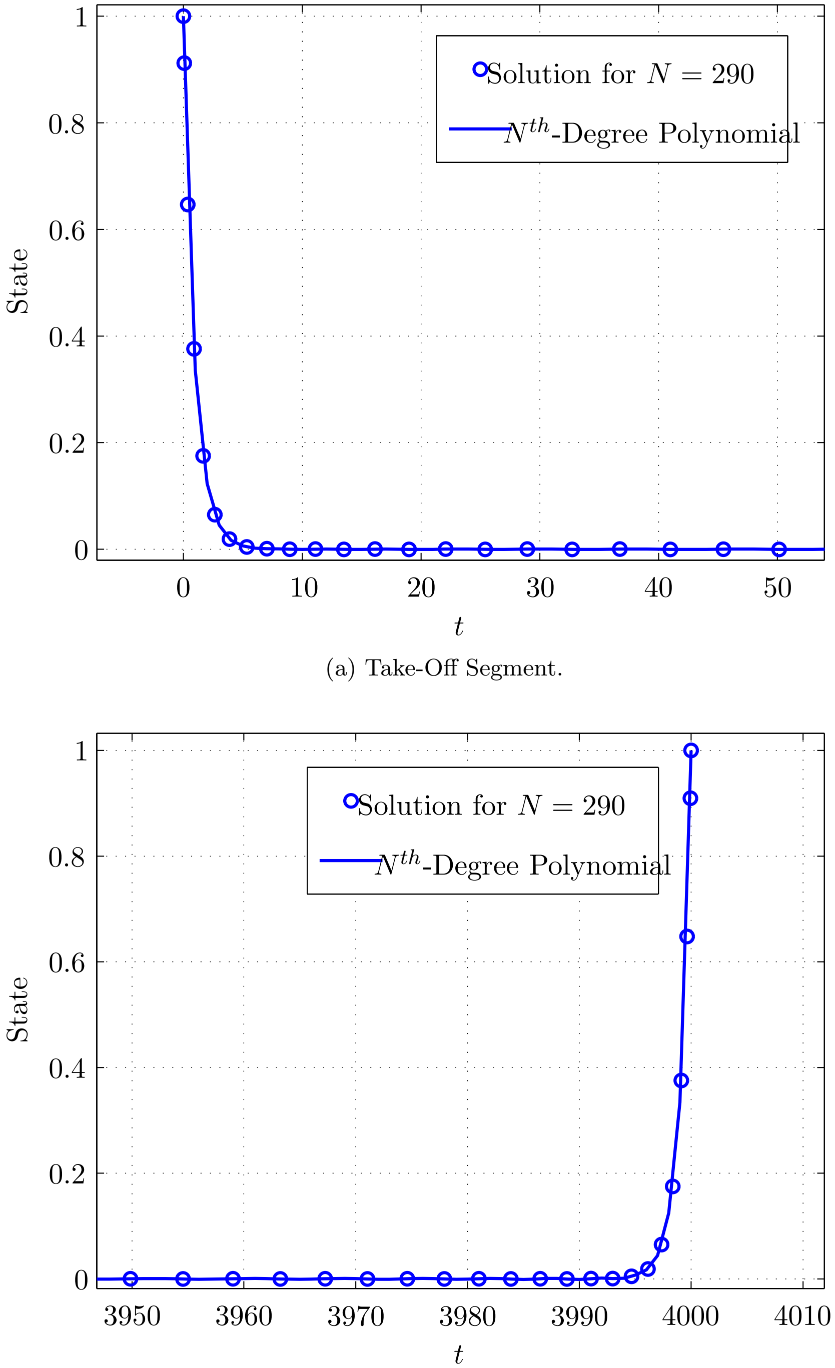


Figure 2: State Solution for Example 1 Using 290 Global Collocation Points Alongside Underlying Lagrange Polynomial Approximation for $t_f = 4000$.

Example 2: Orbit-Raising Problem

Consider the following optimal control problem. Minimize

$$J = t_f \tag{22}$$



(a) Take-Off Segment.

(b) Landing Segment.

Figure 3: Enlargement of State Solution in "Take-Off" and "Landing" Segment for Example 1 Using 290 Global Collocation Points Alongside Underlying Lagrange Polynomial Approximation for $t_f = 4000$.

t_f	CPU Time (s)	Collocation Points	Number of Grids
40	5.12	38	3
200	4.02	72	3
1000	18.72	280	3
2500	25.65	293	3
4000	26.68	290	3

Table 1: Summary of Accuracy and Speed for Example 1 for Different Values of t_f and $\epsilon = 10^{-3}$.

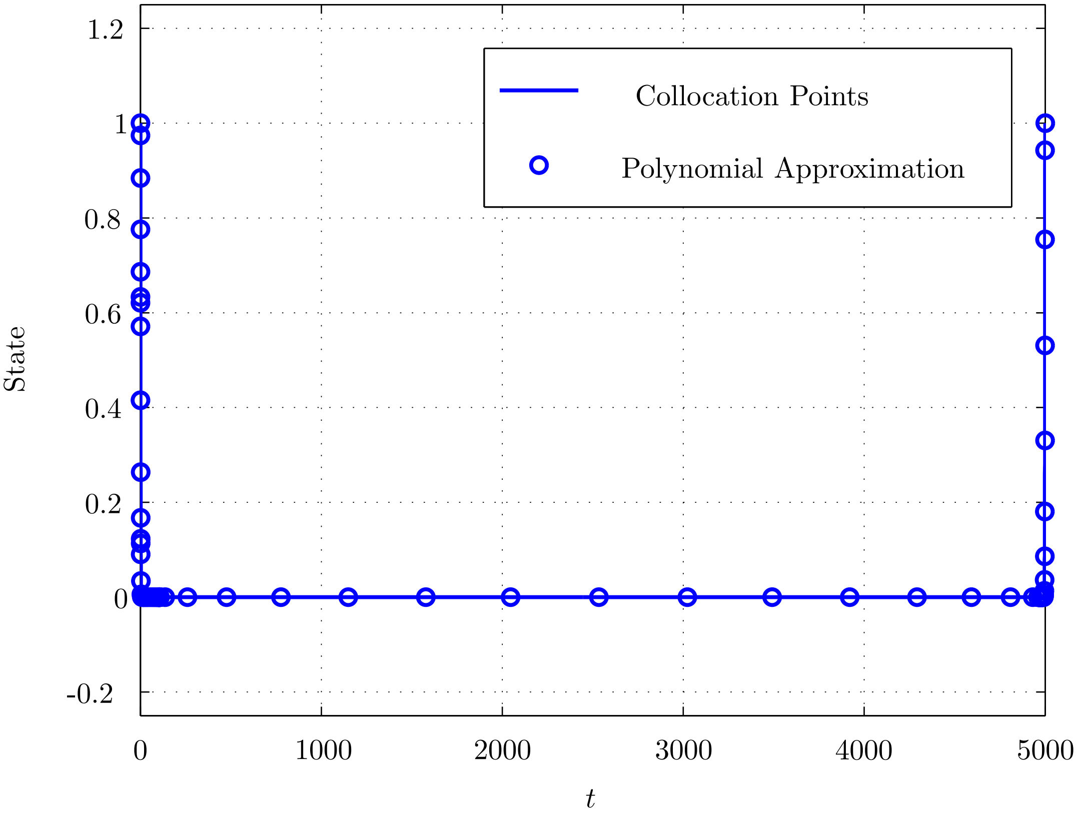
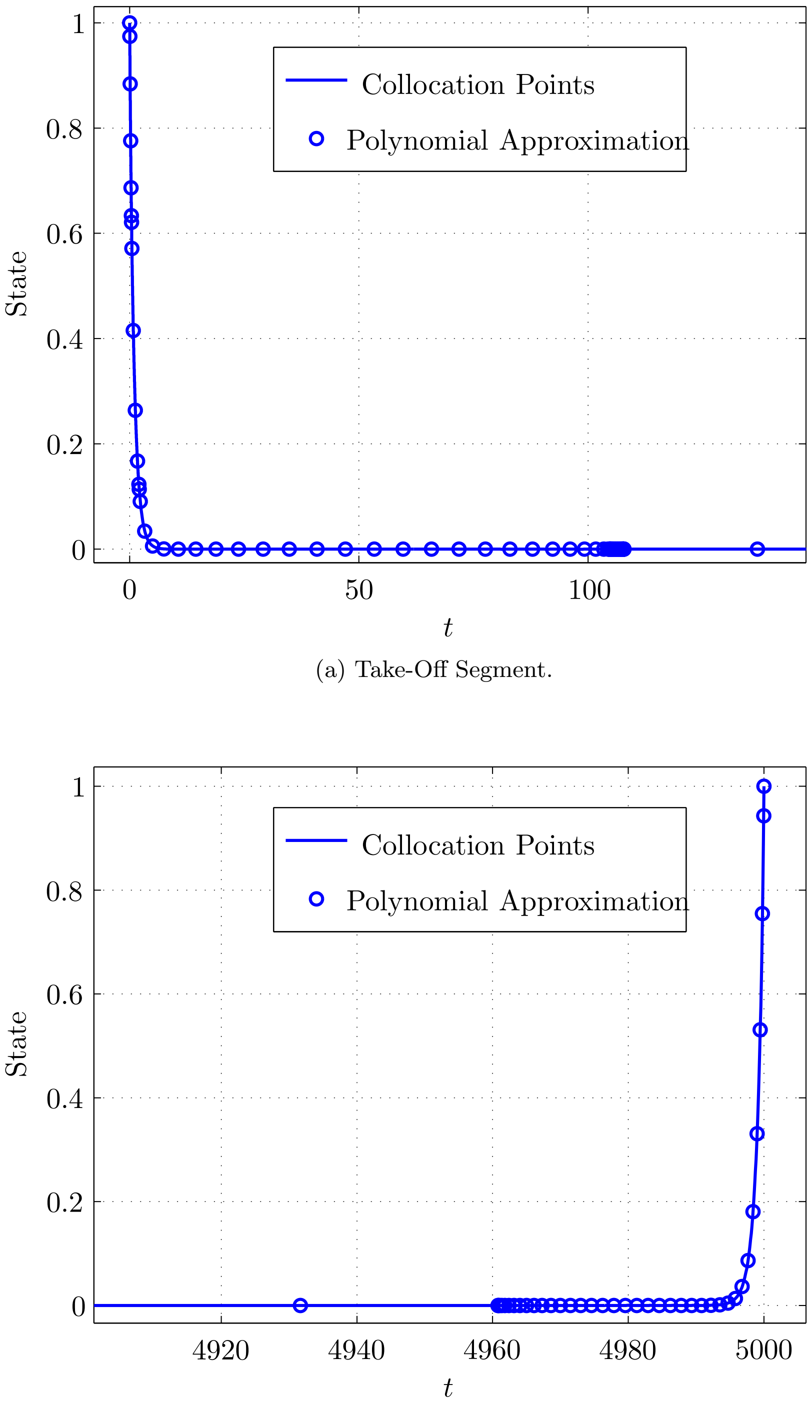


Figure 4: Solution to Example 1 on Final Grid Obtained Using Grid Refinement Algorithm Alongside Lagrange Polynomial Approximation for $t_f = 5000$.



(a) Take-Off Segment.

(b) Landing Segment.

Figure 5: Enlargement of Solution to Example 1 in “Take-Off” and “Landing” Segments on Final Grid Obtained Using Grid Refinement Algorithm Alongside Lagrange Polynomial Approximation for $t_f = 5000$.

t_f	CPU Time (s)	Collocation Points	Segments	Grid Iterations
40	9.79	51	2	6
1000	19.87	65	5	6
2500	27.45	102	6	7
5000	53.85	121	6	9

Table 2: Summary of Accuracy and Speed for Example 1 for Different Values of t_f and $\epsilon = 10^{-3}$.

subject to the dynamic constraints

$$\begin{aligned} \dot{r} &= u \\ \dot{u} &= \frac{v^2}{r} - \frac{\mu}{r^2} + \frac{T}{m} sin(\alpha) \\ \dot{v} &= -\frac{uv}{r} + \frac{T}{m} cos(\alpha) \\ \dot{m} &= -\frac{T}{g_o I_{sp}} \end{aligned} \tag{23}$$

and the boundary conditions

$$\begin{aligned} (r(0), u(0), v(0), m(0)) &= (1, 0, 1, 1) \\ (r(t_f), u(t_f), v(t_f)) &= (1.5, 0, \sqrt{\mu/r(t_f)}) \end{aligned} \tag{24}$$

Fig. 6 displays the control solution and Fig. 7 shows the state solution generated by setting $\rho = 3.5$, using a convergence criterion of 10^{-3}, and initializing the problem with 5 global collocation points. Only the control at the collocation points is shown because there is no underlying polynomial approximation associated with the control. By examining the state and control, the problem appears, visually, to be very smooth. As a result, one might expect that a single segment would suffice. This example demonstrates, however, that by utilizing intelligent segment breaks, the problem can be solved to a particular accuracy using fewer collocation points than is required using global collocation. Table 3 summarizes the results obtained for this example using global collocation (i.e., large values of ρ) and segmentation (in this case, $\rho = 3.5$). It is seen that the algorithm is significantly less computationally efficient using global collocation as compared to when it employs segmentation. In addition, for this example the extrapolation procedure using the global approach did not work as well as in Example 1, requiring two and three extra grid refinements for the 10^{-3} and 10^{-4} cases respectively, implying that the global convergence rate for this problem may be very slow. On the other hand, the algorithm developed in this paper required many fewer collocation points, grid refinements, and computational time. It is noted that the 10^{-3} and 10^{-4} cases have the same solution because the maximum error for both these cases was 9×10^{-5}. The resulting grid consisted of two segments divided at $t = 1.1630$ with 15 collocation points in each segment.

ϵ	ρ	CPU Time (s)	Collocation Points	Segments	Grid Iterations
10^{-2}	Global	68.15	38	1	3
10^{-2}	3.5	4.66	10	2	3
10^{-3}	Global	41.48	79	1	5
10^{-3}	3.5	9.91	30	2	4
10^{-4}	Global	205.30	129	1	6
10^{-4}	3.5	9.91	30	2	4

Table 3: Summary of Accuracy and Speed Using Grid Refinement Algorithm for Example 2 Using Various Accuracy Tolerances, ϵ, and Global/Local Threshold Tolerances, ρ.

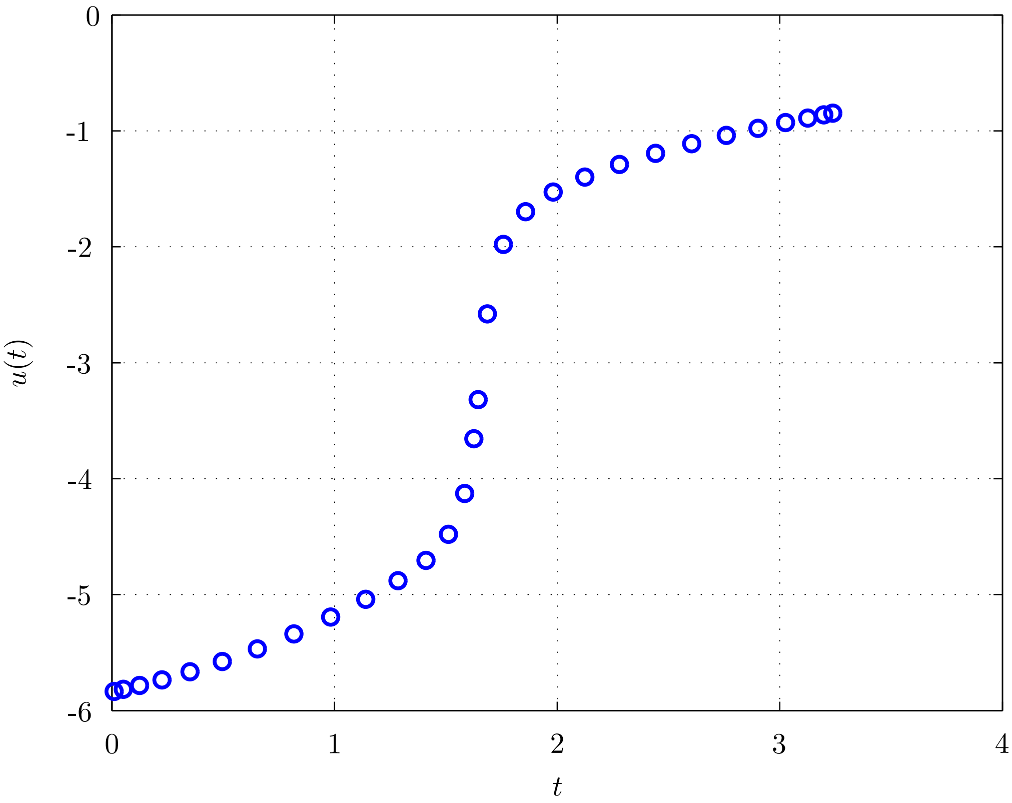


Figure 6: Control vs. Time for Example 2 on Final Grid Obtained Using Grid Refinement Algorithm.

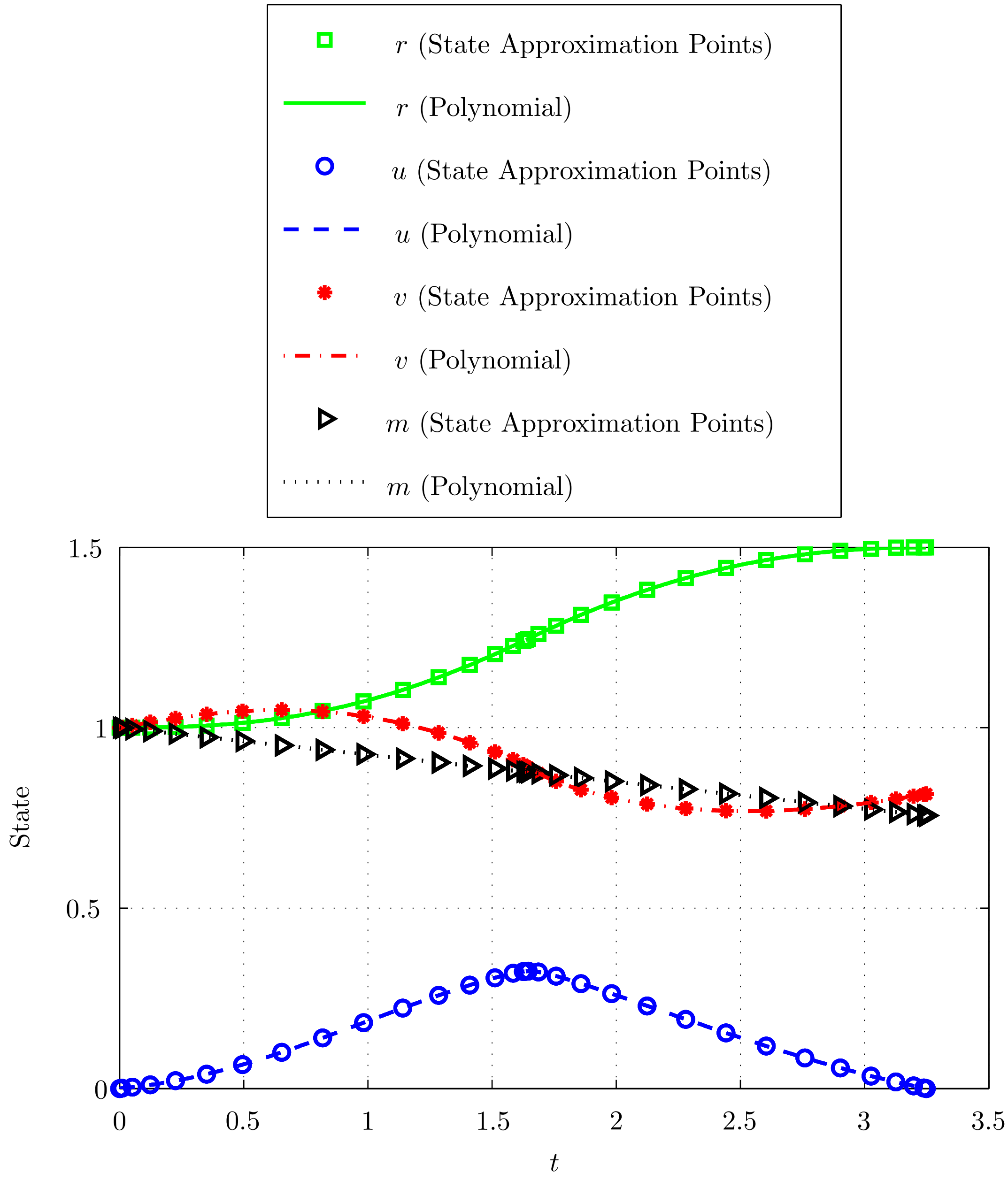


Figure 7: State vs. Time for Example 2 on Final Grid Obtained Using Grid Refinement Algorithm.

Example 3: Moon-Lander Problem

Consider the following optimal control problem. Minimize

$$J = \int_{t_0}^{t_f} u dt \tag{25}$$

subject to

$$\begin{array}{rcl} \dot{h} & = & v \\ \dot{v} & = & -g + u \end{array} \tag{26}$$

where $g = 1.5$, the bounds on the control are $u \in [0, 3]$, initial and final conditions on the state are given as $(h(0), v(0), h(t_f), v(t_f)) = (10, -2, 0, 0)$, and the final time is free. It is known that the optimal control for this example is "bang-bang." Fig. 8 displays the control for this problem with $\epsilon = 10^{-3}$. It is seen that the algorithm converged to five collocation points per segment and determined that the final grid consisted of four segments divided at $t = (1.36, 1.46, 1.56)$. Table 4 displays computational data comparing a global approach with an approach utilizing $\rho = 3.5$. It is seen that, when a highly accurate solution is desired, the global approach is computationally expensive as compared to dividing the problem into segments. In fact, the case using global collocation with $\epsilon = 10^{-4}$ never converged because an unreasonably large number of globally collocated points are required. On the other hand, the algorithm developed in this paper solved the problem much more efficiently even for $\epsilon = 10^{-4}$.

ϵ	ρ	CPU Time (s)	Collocation Points	Segments	Number of Grids
10^{-2}	Global	4.14	35	1	4
10^{-2}	3.5	4.14	20	2	4
10^{-3}	Global	42.03	295	1	6
10^{-3}	3.5	6.37	20	4	6
10^{-4}	Global	—	—	—	—
10^{-4}	3.5	10.31	60	6	9

Table 4: Summary of Accuracy and Speed Using Grid Refinement Algorithm for Example 3 Using Various Accuracy Tolerances, ϵ, and Global/Local Threshold Tolerances, ρ.

Fig. 9 displays the history of the points at which the *state is approximated*[§] for this problem (only the odd history steps are shown because the even steps are used to assess the accuracy of the solution). What is seen is that the node history evolves such that it brackets the discontinuity and places a dense collection of collocation points near the discontinuity and only a few collocation points away from it. It should be noted that ideally a segment break would be placed at the exact location of the discontinuity and only two segments would exist. Practically, this is unattainable because the approximations being used will never, in general, be able to exactly determine the location of a discontinuity.

[§] It is important for the reader to understand that the state in the Gauss pseudospectral method[7–10, 22] is approximated at the LG points plus the endpoints, while the control is approximated at *only the LG points*

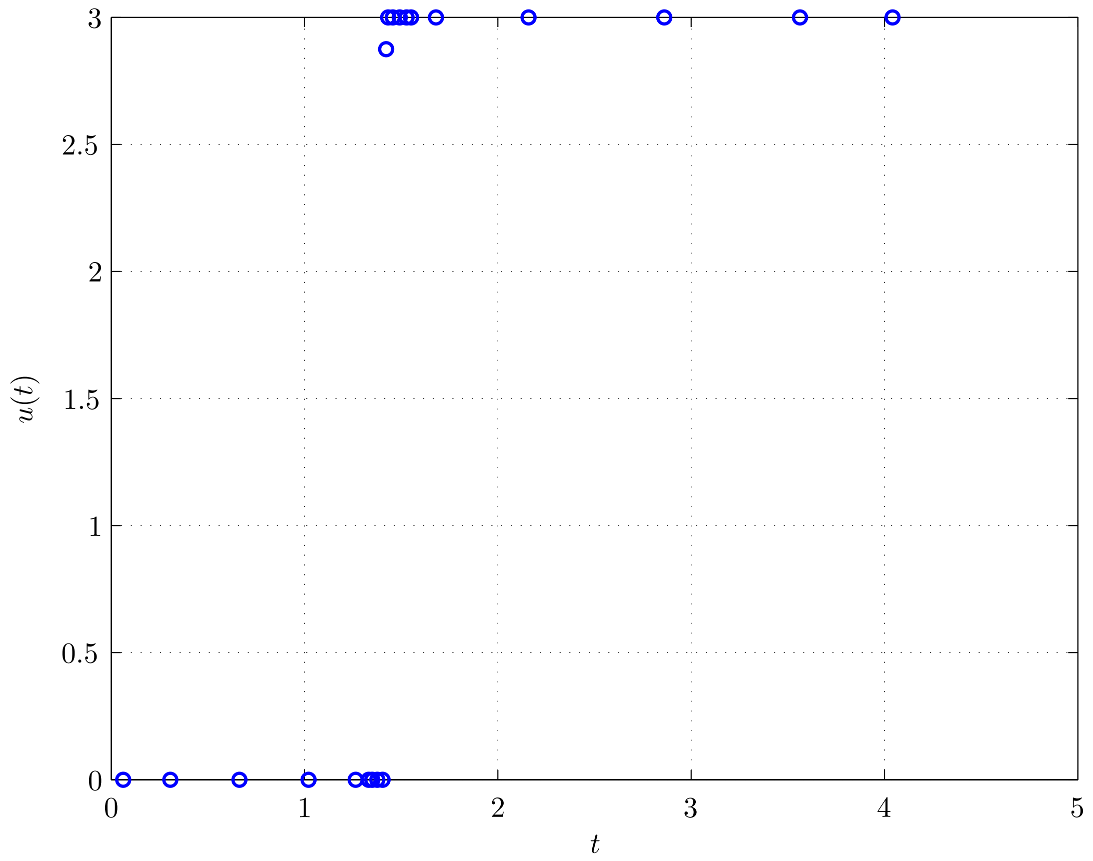


Figure 8: Control vs. Time for Example 3 on Final Grid Obtained Using Grid Refinement Algorithm.

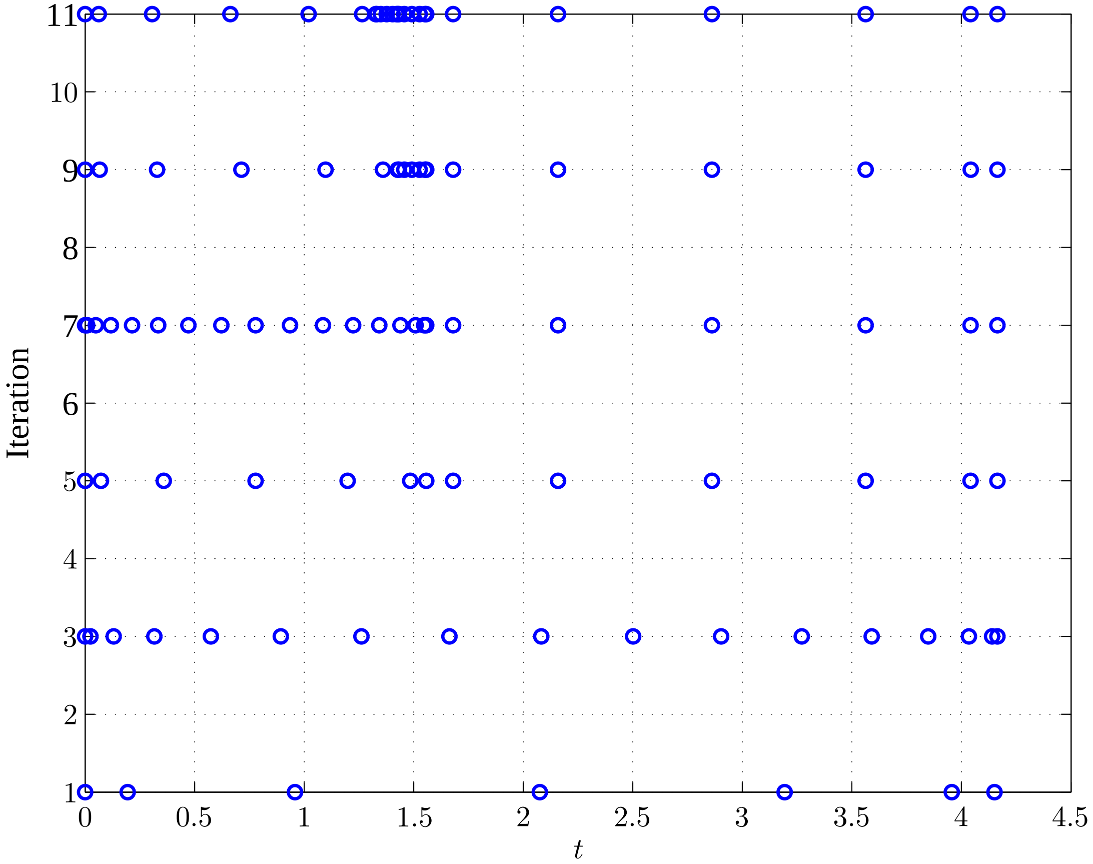


Figure 9: State Approximation Points (i.e., LG Points Plus Endpoints) on the Various Grids for Example 3.

Example 4: Minimum Time-to-Climb of a Supersonic Aircraft

Consider the following optimal control problem. Minimize the cost functional

$$J = t_f \tag{27}$$

subject to the dynamic constraints

$$\begin{aligned} \dot{h} &= v \sin\gamma \\ \dot{E} &= \frac{v}{W}(T - D) \\ \dot{\gamma} &= \frac{g}{v}(\frac{L}{W} - cos\gamma), \end{aligned} \tag{28}$$

the boundary conditions

$$\begin{array}{rclcrcl} h(0) &=& 0 &,& h(t_f) &=& 19995 \text{ m} \\ E(0) &=& 852.6 \text{ m} &,& E(t_f) &=& 24435 \text{ m} \\ \gamma(0) &=& 0 \text{ deg} &,& \gamma(t_f) &=& 0 \text{ deg}, \end{array} \tag{29}$$

and the path constraint $\gamma \leq \pi/4$. It is noted that the vehicle model used in this study is from Ref. 25. The key characteristic of the grid refinement algorithm is captured in Fig. 10, where the optimal flight path angle is shown for the case $\epsilon = 10^{-2}$ with initialization of 20 global nodes. This problem has many interesting features. In the beginning, γ sharply increases from the runway to the path constraint, and then falls off to near zero again. The next rise has the same general shape as the first rapid increase, but it does not hit the path constraint. This problem was divided into four segments ($[5, 5, 15, 15]$ collocation points) connected at $t = 9.22, 18.45, 31.42$. The rapid increase off the runway occurs at approximately $t = 8$ and the path constraint is active from approximately $t = 19$ to $t = 29$. The problem is segmented near these locations, and 15 collocation points are used to capture the active path constraint. Further, the final segment accounts for the majority of the trajectory time and only utilizes 15 collocation points. Although the duration of the segment is long, it is easily approximated because it does not contain reach the inequality path constraint nor has any rapid changes like coming off the runway.

Table 5 displays the computational data for this problem. With regards to computational time, total number of collocation points, and total number of grid iterations, the hybrid approach proved better in all cases.

ϵ	ρ	CPU Time (s)	Collocation Points	Segments	Grid Iterations
10^{-2}	Global	51.75	87	1	7
10^{-2}	3.5	16.66	40	4	3
10^{-3}	Global	542.72	244	1	15
10^{-3}	2.5	99.59	163	6	5
10^{-3}	3.5	46.66	95	5	5

Table 5: Summary of Accuracy and Speed Using Grid Refinement Algorithm for Example 4 Using Various Accuracy Tolerances, ϵ, and Global/Local Threshold Tolerances, ρ.

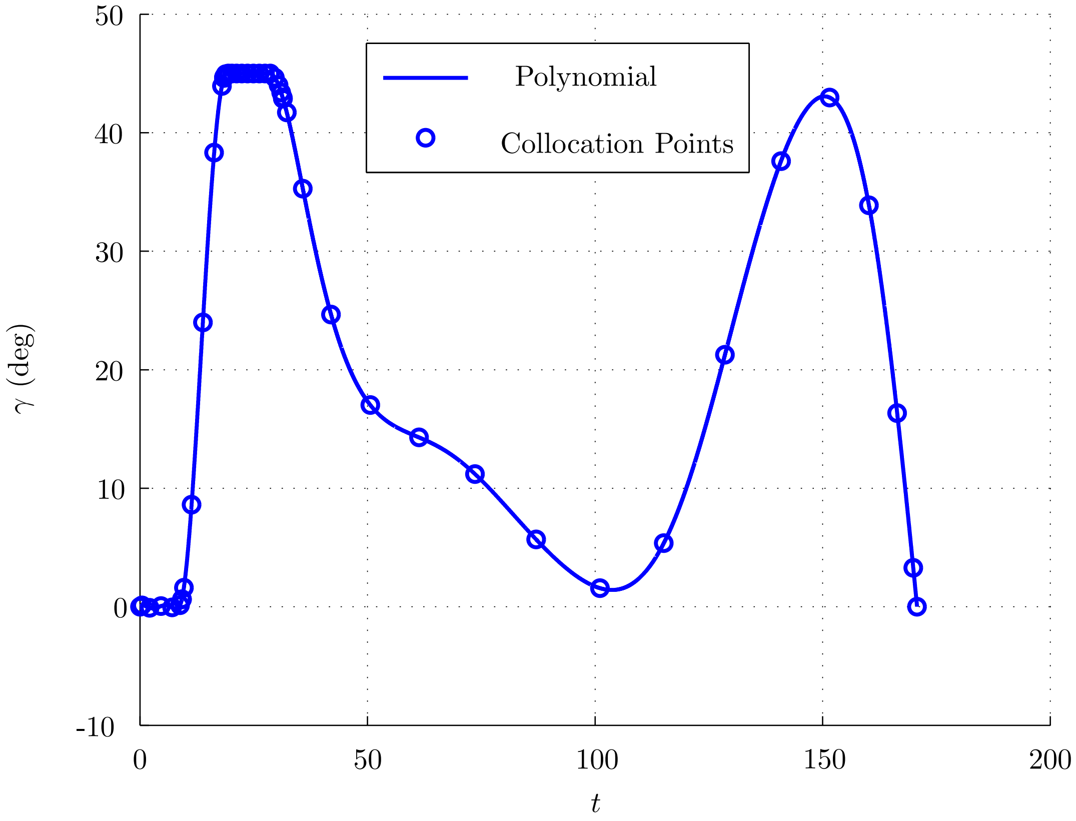


Figure 10: Flight Path Angle vs. Time for Example 4.

VI. Discussion of Results

The results of Section V demonstrate several key features of the algorithm developed in this paper. First, Example 1 shows that, as t_f increases, the take-off and landing (i.e., initial and final) segments are very fast in comparison with the cruise (middle) segment. For small to moderate values of t_f, a global approach was computationally efficient because the degree of the polynomial remained small enough that the problem could be solved quickly using the solution from the previous grid. In addition, the predicted number of collocation points on the next grid was very close the number required to meet the specified tolerance. As a result, very few global grid refinements were required. For larger values of t_f (e.g., $t_f = 5000$), the non-global approach was more computationally efficient than using global collocation. In this case it was found that the node placement was what would be expected, i.e., the algorithm placed a large number of collocation points in the take-off and landing segments (where the action takes place) while putting very few collocation points in the cruise segment.

Examples 2 and 3 demonstrate that the segmented approach is significantly more computationally efficient than the global approach for problems whose solutions are nearly or actually nonsmooth. In particular, the segmented solutions contain many fewer total collocation points as compared with a global approach. Because the solution to Example 3 is nonsmooth (in this case, discontinuous in the control), a global approximation was unable to provide a solution to a specified tolerance of 10^{-4}.

Finally, Example 4 demonstrates how the proposed algorithm treats a problem with active inequality path constraints. Similar to the results of Examples 2 and 3, a segmented approach is more efficient than a global approach. Interestingly, when a segmented approach is used, the segments are divided near the activity/inactivity of the path constraint. In addition, a segment break was placed just after the ascent of the aircraft from the runway. In contrast, the final segment of the trajectory has significantly less interesting behavior and, thus, only one segment with relatively few collocation points was required.

VII. Limitations of the Algorithm

While the algorithm developed in this paper has been found to be successful on a range of problems, it has some limitations that we now describe. First, in the approach developed in this paper, the mesh can only increase in size. For example, inefficiencies are seen in Example 3 where the discontinuity is not accurately captured in the first approximation. Because of these inaccuracies, two segment breaks are placed to the right of the discontinuity before a segment break is placed to the left of the discontinuity. The result of this decomposition is a four segment problem with collocation points being densely located near the discontinuity. Ideally, only two segments would be required, one to the left and another to the right of the discontinuity. In practice, however, one would not expect the discontinuity to be captured perfectly. Instead, one would want the algorithm to divide the problem into *three* segments, where the middle segment would lie very close to the discontinuity and contain a relatively large number of collocation points in the middle segment. The inefficiency of converging to four segments, while not significant in Example 3, may prove to be a computational burden for more complicated applications (e.g., a problem with twenty discontinuities in the control over the trajectory duration). Secondly, the algorithm may fail for a problem with a very high frequency periodic solution. In particular, for such a problem, the algorithm may not divide the problem into segments, but instead simply continue to increase the degree of a global polynomial approximation without bound. Thirdly, by having an assessment step, and then a correction step, the computational burden is doubled for each problem. Combining these steps would result in a significantly more efficient algorithm. Finally, the algorithm assumes a progression of solutions via the NLP which do not have unreasonable defects. If the NLP continuously converges to wrong solutions, i.e., solutions with grid level oscillations, random errors, etc, then the algorithm may never converge.

VIII. Comparison with Other Grid Refinement Methods

The method derived in this paper is distinctly different from other grid refinement techniques. In particular, the goal of many grid refinement techniques is to obtain improve accuracy by increasing the number of collocation points in areas of the trajectory with the largest local errors. An excellent example of a well established local collocation grid refinement method is that given in Ref. 19, where errors are analyzed on a particular grid and the grid is refined based on local errors. If errors are sufficiently large in a particular region, the number of local segments in that region is increased. Another local grid refinement technique is that of Ref. 2, where local smoothness of a solution via interpolation by neighboring points is assessed. In regions where a function is not locally smooth, grid points are added. In the context of pseudospectral methods, Ref. 20 has conducted a study on refining a global pseudospectral solution by analyzing the time derivative of the state or control. First, a suitable global approximation is obtained. The total number of collocation points is then held constant and the problem is segmented at locations where the time derivatives are large. Because the collocation points in a pseudospectral method are denser near the ends of a segment, the algorithm of Ref. 20 provides a denser collection of collocation points at the points of the segment breaks. The algorithm presented in this paper is fundamentally different from both local collocation mesh refinement techniques and the spectral algorithm of Ref. 20 in that the goal of the current algorithm is to determine simultaneously the number of segments and the degree of the polynomial approximation in each segment that results in the best approximation for the solution. Consequently, certain regions of the trajectory may contain many more segments as compared to other regions, and the number of collocation points can vary greatly from segment to segment.

IX. Conclusions

A iterative grid refinement algorithm has been described for solving optimal control problems using pseudospectral methods. A strategy has been devised that determines the locations of segment breaks and the degree of the polynomial approximation required in each segment. The algorithm has been applied to several examples of varying complexity. It has been found that the algorithm produces solutions with equal or better accuracy as compared to using a global pseudospectral approach while utilizing equal or less computation time. The approach developed in this paper has been found to be a viable way to efficiently obtain accurate solutions to optimal control problems using pseudospectral methods.

References

[1]Betts, J. T., *Practical Methods for Optimal Control Using Nonlinear Programming*, SIAM Press, Philadelphia, 2001.

[2]Jain, D. and Tsiotras, P., "Trajectory Optimization Using Multiresolution Techniques," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 5, September-October 2008, pp. 1424–1436.

[3]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.

[4]Herman, A. L., *Improved Collocation Methods with Application to Direct Trajectory Optimization*, Ph.D. thesis, Department of Aerospace Engineering, University of Illinois at Urbana-Champaign, 1995.

[5]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.

[6]Elnagar, G. and Razzaghi, M., "A Collocation-Type Method for Linear Quadratic Optimal Control Problems," *Optimal Control Applications and Methods*, Vol. 18, No. 3, 1998, pp. 227–235.

[7]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.

[8]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.

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

[10]Rao, A. V., Benson, D. A., Darby, C. L., Patterson, M. A., Sanders, I., and Huntington, G. T., *User's Manual for* GPOPS *Version 2.2*, http://gpops.sourceforge.net/, June 2009.

[11]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.

[12]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," Conditionally Accepted to *Computational Optimization and Applications.*

[13]Trefethen, L. N., *Spectral Methods Using MATLAB*, SIAM Press, Philadelphia, 2000.

[14]Davis, P. J. and Rabinowitz, P., *Methods of Numerical Integration*, Academic Press, New York, 1984.

[15]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.

[16]Biegler, L. T. and Zavala, V. M., "Large-Scale Nonlinear Programming Using IPOPT: An Integrating Framework for Enterprise-Wide Optimization," *Computers and Chemical Engineering*, Vol. 33, No. 3, March 2008, pp. 575–582.

[17]Byrd, R. H., Nocedal, J., and Waltz, R. A., "KNITRO: An Integrated Package for Nonlinear Optimization," *Large Scale Nonlinear Optimization*, Springer Verlag, 2006, pp. 35–59.

[18]Huntington, G. T. and Rao, A. V., "A Comparison of Global and Local Collocation Methods for Solving Optimal Control Problems," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 2, March–April 2008, pp. 432–436.

[19]Betts, J. T. and Huffman, W. P., "Mesh Refinement in Direct Transcription Methods for Optimal Control," *Optimal Control Applications and Methods*, Vol. 19, 1998, pp. 1–21.

[20]Gong, Q., Fahroo, F., and Ross, I. M., "Spectral Algorithm for Pseudospectral Methods in Optimal Control," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 3, May-June 2008.

[21]Atkinson, K. E., *An Introduction to Numerical Analysis*, John Wiley and Sons, New York, NY, 1989.

[22]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,* Accepted for Publication, June 2009.

[23]Gill, P. E., Murray, W., and Saunders, M. A., *User's Guide for SNOPT Version 7: Software for Large Scale Nonlinear Programming*, February 2006.

[24] Rao, A. V. and Mease, K. D., "Eigenvector Approximate Dichotomic Basis Method for solving Hyper-Sensitive optimal Control Problems," *Optimal Control Applications and Methods*, Vol. 21, No. 1, January–February 2000, pp. 1–19.

[25] Seywald, H. and Kumar, R. R., "Finite Difference Scheme for Automatic Costate Calculation," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 4, 1996, pp. 231–239.