JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 27, No. 3, May–June 2004

Pseudospectral Knotting Methods for Solving Optimal Control Problems

I. Michael Ross* and Fariba Fahroo†
Naval Postgraduate School, Monterey, California 93943

A class of computational methods for solving a wide variety of optimal control problems is presented; these problems include nonsmooth, nonlinear, switched optimal control problems, as well as standard multiphase problems. Methods are based on pseudospectral approximations of the differential constraints that are assumed to be given in the form of controlled differential inclusions including the usual vector field and differential-algebraic forms. Discontinuities and switches in states, controls, cost functional, dynamic constraints, and various other mappings associated with the generalized Bolza problem are allowed by the concept of pseudospectral (PS) knots. Information across switches and corners is passed in the form of discrete event conditions localized at the PS knots. The optimal control problem is approximated to a structured sparse mathematical programming problem. The discretized problem is solved using off-the-shelf solvers that include sequential quadratic programming and interior point methods. Two examples that demonstrate the concept of hard and soft knots are presented.

Introduction

OVER the last few years, pseudospectral (PS) methods, and in particular the Legendre PS method, have been extensively used to solve a broad class of optimal control problems arising in the trajectory optimization and real-time control of systems governed by ordinary differential equations. Examples range from low-thrust orbit transfers,[1] impulsive orbit transfers,[2] pick and place maneuver of robots,[3] solar sail trajectory optimization,[4,5] ascent guidance,[6,7] reentry trajectory design,[8,9] spacecraft attitude control,[10,11] tethered satellite system control,[12] and many more. Many of these applications of PS methods have been facilitated by various versions of the software package DIDO[13] that often have certain implementations of concepts that are not published in a peer-reviewed journal. The concept of PS knots is one such example. In this paper, we formally introduce the concept of PS knots and show how this simple idea extends a standard direct PS method[14,15] to solving a large class of optimal control problems that include the problems of switches and discontinuities frequently encountered in engineering applications. In one sense, this paper brings together the concepts introduced in Refs. 16 and 17. In addition, we solve a hybrid optimal control problem and demonstrate nonsmooth concepts using a neoclassic optimal control problem introduced by Clarke.[18] Thus, the concept of phases, frequently employed in standard methods,[19,20] is automatically subsumed in this formulation. Such problem formulations frequently arise in the mission design of interplanetary spacecraft trajectories[5] and multistage launch vehicle trajectory optimization.[7]

PS methods along with Galerkin and tau methods are major examples of spectral methods, which have been used extensively in computational fluid mechanics.[21,22] In fact, the driver for the development of spectral methods was problems in fluid dynamics and not optimal control. The first introduction of spectral methods (Legendre–tau) for solving optimal control problems (governed by partial differential equations) was initiated in Ref. 23 for solving problems in active noise control. This work was followed by the application of the PS methods in Refs. 14 and 24 for solving nonlinear control problems governed by ordinary differential equations. Since these initial works, PS methods have been used to solve a wide range of problems, as mentioned earlier. PS methods have also been extended to solve problems governed by differential inclusions,[15] differential-algebraic equations,[25] and higher-order dynamics,[26] as well as differentially flat systems.[27]

Standard PS approximations[14,28] in optimal control are based on expanding the underlying functions in terms of interpolating polynomials, which interpolate these functions at some specially chosen nodes. These nodes are zeros of orthogonal polynomials (or their derivatives) such as Legendre polynomials (Legendre–Gauss points) or Chebyshev polynomials (Chebyshev points). What distinguishes these methods is the choice of these nodes, subsequently the expression for the Lagrange interpolating polynomials. These methods are quite efficient and more accurate than the traditional collocation methods[19,20] in solving smooth optimal control problems, but their use in solving nonsmooth and hybrid problems, that is, problems with switches, can cause major difficulties. For example, interior point constraints cannot be accurately handled by the smooth pseudospectral method because the location of the point may not be at the preallocated Gauss node. Adding more nodes for mesh refinements could lead to inefficiencies and ill conditioning of the discretized problem. Furthermore, many practical problems include empirical models based on table lookups, which are often nonsmooth data.[20] Also, jump discontinuities in states (such as those encountered in the trajectory optimization of multistage launch vehicles) cannot be handled by these methods. Even for problems with smooth data, the solution may be nonsmooth.[18] In this case, PS methods exhibit the well-known Gibbs phenomenon (see Ref. 29) resulting from the approximation of a nonsmooth function by a finite number of smooth functions. These difficulties are fundamentally due to the use of global orthogonal polynomials and nodal points that have a predetermined distribution. This distribution of nodal points yields optimal interpolation, but offers no choice in the selection of the points. Some of these difficulties can be overcome by the concept of spectral patching,[30] but a strict application of patching limits the method to only a reallocation of the nodes.

All of the mentioned difficulties are eliminated by the concept of PS knots. PS knots essentially generalize the spectral patching method by exchanging information across the patches in the form of event conditions associated with the optimal control problem. We introduce the concepts of hard and soft knots for a method based on the Legendre PS method, but it is trivially applicable for other PS methods as well. In between these knots, the problem is discretized at the Legendre–Gauss–Lobatto (LGL) nodes. The discretization of the dynamic constraints is achieved by a differentiation operator that naturally allows for one-sided stencils[29] at the knots, whereas

Received 28 June 2003; revision received 21 September 2003; accepted for publication 22 September 2003.

*Associate Professor, Department of Mechanical and Astronautical Engineering, Code AA/Ro; imross@nps.navy.mil. Associate Fellow AIAA.
†Associate Professor, Department of Applied Mathematics, Code MA/Ff; ffahroo@nps.navy.mil. Senior Member AIAA.

the integral associated with the cost function is approximated by a Gauss quadrature. Information to nodes across the knots is passed in the form of event or switching conditions localized at the knots. The concept of knots can be best illustrated by considering a class of prototype problems based on a single-switching condition discussed in the next section.

Prototype Problems

The class of problems that can be addressed by PS knotting methods is nonsmooth hybrid optimal control problems[18,31] that subsume the notion of multiphase optimal control problems.[19,20] A formal mathematical formulation of such problems requires significant mathematical machinery (for example, Refs. 32 and 33), which we circumvent for the purposes of engineering applications. The route we adopt is a formalization of some intuitive concepts. In this regard, we employ the concept of a multifunction.[18,32] Whereas a function, $f:\mathbb{R}\to\mathbb{R}$, is a point-to-point map, a multifunction is a point-to-set map, denoted as $f:\mathbb{R}\twoheadrightarrow\mathbb{R}$, where the symbol $\twoheadrightarrow$ means that the image of f is a set in $\mathbb{R}$. The notations[18,32] $\rightsquigarrow$ and $\rightrightarrows$ are also used to represent a point-to-set map, but we prefer the more evocative notation $\twoheadrightarrow$ used by Sussmann.[33] In many instances, a multifunction is essentially a formalization of concepts frequently used in engineering without much fanfare. For example, the Heaviside step function $H(\tau)$ is frequently plotted as shown in Fig. 1, but under the concept of a function, $H(0)$ is argued to be either 0, 1, 1/2, or irrelevant. This is done to conform to the notion that the image of a function be a point. This conceptual difficulty is overcome rather simply by a set-valued definition of the step function as

$$H(\tau)=\begin{cases}0 & \tau\le 0\\ [0,\,1] & \tau=0\\ 1 & \tau\ge 1\end{cases} \tag{1}$$

so that the intuitive multivalued nature of the function at $\tau=0$ is embraced while recognizing that nonsmoothness typically occur at discrete points, that is, over zero measure. Thus, over $\tau\in\mathbb{R}\backslash\{0\}$, the step multifunction conforms with the standard definition of the step function. One of the main mathematical reasons for viewing nonsmoothness as multifunctions is that a consistent calculus can be developed.[18,32,33] For example, the generalized derivative of the absolute-value function, $\tau\mapsto|\tau|$, is the signum multifunction: $\mathrm{sgn}(\tau)=2H(\tau)-1$. At the origin (which may be the only point of interest because it is the minimum), the generalized derivative of the absolute-value function is the closed interval, $[-1,\,1]$, that is, a set. Because $0\in[-1,\,1]$, it is apparent that the vanishing of the ordinary derivative for the extremum of a function ($0=\partial f/\partial\tau$) is generalized to the notion that zero be an element of the generalized derivative ($0\in\partial f$). The idea of a multifunction is applicable to concepts well beyond generalizing jumps. For example, a control set $\mathbb{U}\subset\mathbb{R}^{N_u}$ can be viewed as a multifunction, $\mathbb{U}(\cdot):[\tau_0,\,\tau_f]\twoheadrightarrow\mathbb{R}^{N_u}$ with a control function $\boldsymbol{u}(\cdot):[\tau_0,\,\tau_f]\to\mathbb{R}^{N_u}$ being a selection from this set. Throughout this paper, we will use the notation $N_{(\cdot)}$ to mean some element of $\mathbb{N}$, the set of natural numbers. To further facilitate the concepts used, let $\mathcal{X}$ be the space of piecewise differentiable bounded functions. In a formal setting, $\mathcal{X}$ must be replaced by the space of functions that are Lipschitz continuous almost everywhere (see Ref. 32). As noted before, we avoid such formalities to facilitate a wider exposition to the concepts delineated in this paper.

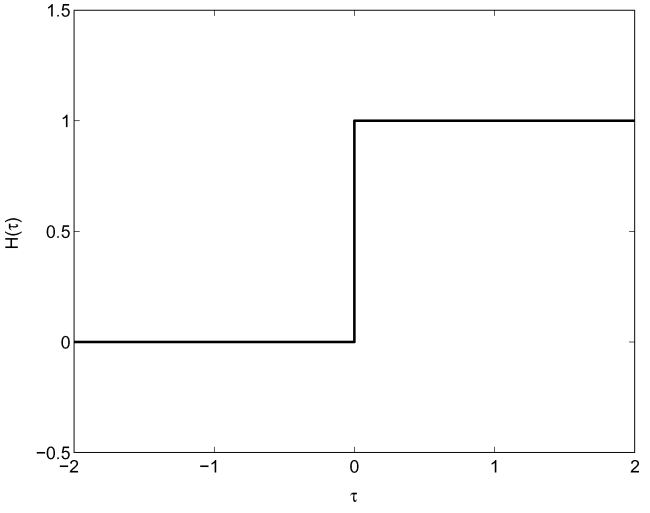

Fig. 1 Step function as a set-valued map; at $\tau=0$, $H(\tau)$ is set [0, 1].

Now, consider an arbitrary point $\tau_e\in(\tau_0,\,\tau_f)$, where $[\tau_0,\,\tau_f]=I\subset\mathbb{R}$ is a time interval of interest (which may be fixed or free). Let $\tau\mapsto\boldsymbol{x}\in\mathbb{R}^{N_x}$ be any function from the space $\mathcal{X}$, so that τ_e is a point of potential interest. Let

$$\boldsymbol{x}_e^-=\lim_{\tau\uparrow\tau_e}\boldsymbol{x}(\tau),\qquad \boldsymbol{x}_e^+=\lim_{\tau\downarrow\tau_e}\boldsymbol{x}(\tau) \tag{2}$$

Suppose that N_x is the same before and after τ_e. (Switches in N_x are also allowed as will be apparent shortly.) We define $\boldsymbol{x}(\tau_e)$ to be the set

$$\boldsymbol{x}(\tau_e)=\{\boldsymbol{y}:\boldsymbol{y}=\alpha\boldsymbol{x}^-+(1-\alpha)\boldsymbol{x}^+,\,\alpha\in[0,\,1]\} \tag{3}$$

This generalized definition of the value of a function implies that, if $\boldsymbol{x}(\cdot)$ is continuous at τ_e, then $\boldsymbol{x}(\tau_e)$ conforms to the usual notion of the value of a function as the right-hand side of Eq. (3) degenerates to a point. Thus, the interval I may be arbitrarily divided into two closed subintervals $I^1=[\tau_0,\,\tau_e]$ and $I^2=[\tau_e,\,\tau_f]$ so that $I^1\cap I^2=\{\tau_e\}$ is of no consequence if all of the generalized notions reduce to the familiar ones at τ_e. The observation is made that $I^1\cap I^2$ does not equal the empty set but does equal the singleton $\{\tau_e\}$ will be explicitly exploited in the discretization method and is central to the concept of knots.

Let $\boldsymbol{x}\in\mathbb{R}^{N_x}$ and $\boldsymbol{u}\in\mathbb{R}^{N_u}$ be the state and control variables, respectively. We define prototype problems based on a single event time τ_e. We call τ_e an event time if it satisfies the discrete event condition

$$\boldsymbol{e}^L\le\boldsymbol{e}\left(\boldsymbol{x}_0,\boldsymbol{x}_e^-,\boldsymbol{x}_e^+,\boldsymbol{x}_f;\boldsymbol{u}_e^-,\boldsymbol{u}_e^+;\tau_0,\tau_e,\tau_f;\boldsymbol{p}\right)\le\boldsymbol{e}^U \tag{4}$$

where $\boldsymbol{e}^L\in\mathbb{R}^{N_e}$ and $\boldsymbol{e}^U\in\mathbb{R}^{N_e}$ are the lower and upper bounds on the value of the function $\boldsymbol{e}$,

$$\boldsymbol{e}:\mathbb{R}^{N_{x^1}}\times\mathbb{R}^{N_{x^1}}\times\mathbb{R}^{N_{x^2}}\times\mathbb{R}^{N_{x^2}}\times\mathbb{R}^{N_{u^1}}\times\mathbb{R}^{N_{u^2}}\times\mathbb{R}\times\mathbb{R}\times\mathbb{R}\times\mathbb{P}\to\mathbb{R}^{N_e} \tag{5}$$

where N_e is the dimension of the event function, $\boldsymbol{x}_0=\boldsymbol{x}(\tau_0)\in\mathbb{R}^{N_{x^1}}$, $\boldsymbol{x}_f=\boldsymbol{x}(\tau_f)\in\mathbb{R}^{N_{x^2}}$, $\boldsymbol{u}_e^-$ and $\boldsymbol{u}_e^+$ are defined similar to Eq. (2), and $\mathbb{P}$ is the space of parameters that may be real, integers, or some combination of real and integer space. The set of all points $(\boldsymbol{x}_0,\boldsymbol{x}_e^-,\boldsymbol{x}_e^+,\boldsymbol{x}_f;\boldsymbol{u}_e^-,\boldsymbol{u}_e^+;\tau_0,\tau_e,\tau_f;\boldsymbol{p})$ satisfying Eq. (4) will be denoted as the event set $\mathbb{E}$.

Several points are worth noting in the representation of $\mathbb{E}$ given by Eq. (4). First, no distinction is made between equalities and inequalities; an equality is simply obtained by setting the lower bound equal to its upper bound. Second, no distinction is made between the initial point, the interior point, and the final point. In this sense the boundary conditions are not necessarily split, such as, for example, in a hybrid control formulation[31] where the initial and final conditions are given as

$$\boldsymbol{e}_0^L\le\boldsymbol{e}_0(\boldsymbol{x}_0,\tau_0)\le\boldsymbol{e}_0^U \tag{6}$$

$$\boldsymbol{e}_f^L\le\boldsymbol{e}_f(\boldsymbol{x}_f,\tau_f)\le\boldsymbol{e}_f^U \tag{7}$$

whereas the interior point condition is stipulated as

$$\boldsymbol{e}_i^L\le\boldsymbol{e}_i\left(\boldsymbol{x}_e^-,\boldsymbol{x}_e^+,\tau_e\right)\le\boldsymbol{e}_i^U \tag{8}$$

Such a formulation is subsumed in Eq. (4) by simply letting

$$\boldsymbol{e}^L = \begin{pmatrix} \boldsymbol{e}_0^L \\ \boldsymbol{e}_i^L \\ \boldsymbol{e}_f^L \end{pmatrix} \qquad \boldsymbol{e}^U = \begin{pmatrix} \boldsymbol{e}_0^U \\ \boldsymbol{e}_i^U \\ \boldsymbol{e}_f^U \end{pmatrix}$$

$$\boldsymbol{e}\left(\boldsymbol{x}_0, \boldsymbol{x}_e^-, \boldsymbol{x}_e^+, \boldsymbol{x}_f; \tau_0, \tau_e, \tau_f\right) = \begin{bmatrix} \boldsymbol{e}_0(\boldsymbol{x}_0, \tau_0) \\ \boldsymbol{e}_i\left(\boldsymbol{x}_e^-, \boldsymbol{x}_e^+, \tau_e\right) \\ \boldsymbol{e}_f(\boldsymbol{x}_f, \tau_f) \end{bmatrix}$$

Third, the initial time, the final time, and the interior event time are all allowed to be fixed or free, and no explicit distinction is made between time-fixed and time-free problems. For example, if the initial time τ_0 is fixed, for example, at 0, then an event constraint function, for example, e_1, the first component of $\boldsymbol{e}$ is simply defined as τ_0 and its lower bound and upper bound (e_1^L and e_1^U) are set to 0. Having the initial time be a free variable is important in some applications such as those arising in finding optimal launch windows in conjunction with optimizing space trajectories. In other words, Eq. (4) is a strikingly simple representation of the event set $\mathbb{E}$, while being substantially general in allowing fairly complex formulations of boundary conditions and interior point constraints including discontinuities in the state and control variables and interdependence of point constraints beyond just the adjacent event times. Such a parallel formulation is not performed just for the sake of greater mathematical generality; rather, it is quite crucial for clearly formulating practical engineering problems such as those arising in space mission design involving continuous and discontinuous variables as illustrated, for example, in Ref. 5. Fourth, we note that the possibly distinct values of the left- and right-hand limits of the state and control variables are respected, which, as will be apparent later, are explicitly taken into account by the concept of PS knots.

Similar to the representation of the event function, we can define an event cost function E,

$$E : \mathbb{R}^{N_{x^1}} \times \mathbb{R}^{N_{x^1}} \times \mathbb{R}^{N_{x^2}} \times \mathbb{R}^{N_{x^2}} \times \mathbb{R}^{N_{u^1}} \times \mathbb{R}^{N_{u^2}} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{P} \to \mathbb{R} \tag{9}$$

which is defined over the same domain as the event function. The running cost F, that is, the integrand in the Bolza cost functional,

$$J[\boldsymbol{x}(\cdot), \boldsymbol{u}(\cdot), \tau_0, \tau_e, \tau_f, \boldsymbol{p}] = E\left(\boldsymbol{x}_0, \boldsymbol{x}_e^-, \boldsymbol{x}_e^+, \boldsymbol{x}_f; \boldsymbol{u}_e^-, \boldsymbol{u}_e^+; \tau_0, \tau_e, \tau_f; \boldsymbol{p}\right) + \int_{\tau_0}^{\tau_f} F(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau; \boldsymbol{p})\, \mathrm{d}\tau \tag{10}$$

can have a representation given by

$$F(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau, \boldsymbol{p}) = \begin{cases} F^1(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau; \boldsymbol{p}) & \tau \in I^1 \\ F^2(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau; \boldsymbol{p}) & \tau \in I^2 \end{cases} \tag{11}$$

where $F^i : \mathbb{R}^{N_{x^i}} \times \mathbb{R}^{N_{u^i}} \times \mathbb{R} \times \mathbb{P} \to \mathbb{R}$, $i = 1, 2$, so that F is allowed to have the possibility

$$\lim_{\tau \uparrow \tau_e} F(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau, \boldsymbol{p}) \neq \lim_{\tau \downarrow \tau_e} F(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau, \boldsymbol{p}) \tag{12}$$

Note that F is possibly multivalued at the point τ_e and is, hence, a multifunction. Similarly, we define the path constraint to be mixed with state and control variables,

$$\boldsymbol{h}^L \leq \boldsymbol{h}(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau; \boldsymbol{p}) \leq \boldsymbol{h}^U \tag{13}$$

where $\boldsymbol{h}^L$ and $\boldsymbol{h}^U$ are the lower and upper bounds on the values of the function and $\boldsymbol{h}$ is

$$\boldsymbol{h}(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau, \boldsymbol{p}) = \begin{cases} \boldsymbol{h}^1(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau; \boldsymbol{p}) & \tau \in I^1 \\ \boldsymbol{h}^2(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau; \boldsymbol{p}) & \tau \in I^2 \end{cases} \tag{14}$$

where $\boldsymbol{h}^i : \mathbb{R}^{N_{x^i}} \times \mathbb{R}^{N_{u^i}} \times \mathbb{R} \times \mathbb{P} \to \mathbb{R}^{N_{h^i}}$, $i = 1, 2$. If $N_{h^1} = N_{h^2}$, we allow the possibility

$$\lim_{\tau \uparrow \tau_e} \boldsymbol{h}(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau; \boldsymbol{p}) \neq \lim_{\tau \downarrow \tau_e} \boldsymbol{h}(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau; \boldsymbol{p}) \tag{15}$$

Finally, we allow the differential constraints to be specified in terms of a nonsmooth hybrid controlled differential inclusion that may be specified functionally as

$$\boldsymbol{f}^L \leq \boldsymbol{f}(\dot{\boldsymbol{x}}(\tau), \boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau; \boldsymbol{p}) \leq \boldsymbol{f}^U \tag{16}$$

where $\boldsymbol{f}^L$ and $\boldsymbol{f}^U$ are the lower and upper bounds on the values of the function $\boldsymbol{f}$ represented by

$$\boldsymbol{f}(\dot{\boldsymbol{x}}(\tau), \boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau; \boldsymbol{p}) = \begin{cases} \boldsymbol{f}^1(\dot{\boldsymbol{x}}(\tau), \boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau; \boldsymbol{p}) & \tau \in I^1 \\ \boldsymbol{f}^2(\dot{\boldsymbol{x}}(\tau), \boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau; \boldsymbol{p}) & \tau \in I^2 \end{cases} \tag{17}$$

where $\boldsymbol{f}^i : \mathbb{R}^{N_{x^i}} \times \mathbb{R}^{N_{x^i}} \times \mathbb{R}^{N_{u^i}} \times \mathbb{R} \times \mathbb{P} \to \mathbb{R}^{N_{f^i}}$, $i = 1, 2$. In our generalized formulation of differential constraints we may also have (if $N_{x^1} = N_{x^2}$)

$$\lim_{\tau \uparrow \tau_e} \dot{\boldsymbol{x}}(\tau) \neq \lim_{\tau \downarrow \tau_e} \dot{\boldsymbol{x}}(\tau) \tag{18}$$

in addition to switches in the function $\boldsymbol{f}$ itself. Furthermore, unlike an ordinary differential inclusion, a controlled differential inclusion is more common in engineering applications. For example, for $|u_1| \leq 1$, the system of differential equations

$$\dot{x}_1 = u_1 \tag{19}$$

$$\dot{x}_2 = x_1 u_1 \cos u_2 + x_3 u_2^3 \tag{20}$$

$$\dot{x}_3 = x_2 \sinh u_1 \tag{21}$$

can be readily written as a controlled differential inclusion,

$$-1 \leq \dot{x}_1 \leq 1 \tag{22}$$

$$\dot{x}_2 = x_1 u_1 \cos u_2 + x_3 u_2^3 \tag{23}$$

$$\dot{x}_3 = x_2 \sinh u_1 \tag{24}$$

without losing information by this partial hodograph transformation. In some situations, a full passage to ordinary differential inclusions is accompanied by a loss in information.[33] In other words, a controlled differential inclusion allows partial elimination of the control variables, whereas an ordinary differential inclusion demands that all control variables be eliminated: a severe requirement for dynamic systems that are not control affine. In recent years, many new formulations of dynamics are directly given in terms of differential inclusions and not through control elimination. See Ref. 34 and the references contained therein for a vast array of examples in rigid-body dynamics, Ref. 35 for an excellent survey on numerical methods for differential inclusions, and Ref. 36 for a perspective on inclusions and other transformations.

Note that our method applies to differential inclusions that may be functionally specified in ways other than differential inequalities.[36] In addition, various scenarios can happen both physically and in terms of modeling across an event. For example, over one segment, the dynamics of the problem may be given in terms of an ordinary differential inclusion,

$$\boldsymbol{f}^{L^1} \leq \boldsymbol{f}^1(\dot{\boldsymbol{x}}(\tau), \boldsymbol{x}(\tau), \tau) \leq \boldsymbol{f}^{U^1} \tag{25}$$

that is, absence of control, whereas in another segment it may be defined in terms of a controlled differential algebraic equation,

$$\boldsymbol{f}^2(\dot{\boldsymbol{x}}(\tau), \boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau) = \boldsymbol{0} \tag{26}$$

that is, $\boldsymbol{f}^{L^2} = \boldsymbol{f}^{U^2} = \boldsymbol{0}$. As noted earlier, such formulations are motivated not just for the sake of mathematical generality but because practical engineering problems indeed exhibit such complexity. For example, in the case of a particular example of a low-thrust trajectory with terminal aerocapture,[37] the exoatmospheric segment has only thrusting capability and no aerodynamic controls, whereas the endoatmospheric segment has no thrusting capability. In addition,

the dimensions of the states and controls are different across the aerocapture event. Other possibilities such as higher-order differential constraints[26] over selected segments are also allowed, but we omit these descriptions for the sake of clarity in describing a prototype problem.

We now define our prototype optimal control problem as follows: Determine the possibly discontinuous switched state-control function pair $[\tau_0, \tau_f] \ni \tau \mapsto \{\boldsymbol{x}, \boldsymbol{u}\}$; the optimal event times $\tau_0 \in \mathbb{R}$, $\tau_e \in (\tau_0, \tau_f)$ and $\tau_f \in \mathbb{R}$; and the parameter $\boldsymbol{p} \in \mathbb{P}$ that minimize the generalized Bolza cost functional given by Eq. (10), subject to the differential inclusions of Eq. (16), the mixed-state control path constraints given by Eq. (13), and the event constraints Eq. (4).

This problem formulation is deceptively simple and is superficially similar to a somewhat standard Bolza problem except that we now allow the functions $F, \boldsymbol{f}$, and $\boldsymbol{h}$ to have nonsmooth and hybrid characteristics that include possible switchings in their dimensions across an event. Note that, even if $\boldsymbol{h}$ is smooth, it automatically generates nonsmooth trajectories[32] because of the mixed nature of the path constraints. That is, even with smooth data, optimal trajectories can be nonsmooth. Note in our problem formulation that at τ_e one or more controls maybe a Dirac delta function that causes a jump in the state variables. The effect of this Dirac function and its cost can be modeled in terms of the event pair $\{E, \boldsymbol{e}\}$. An example of this may be found in Ref. 2. Finally, it is apparent that the prototype problem formulation easily extends to more than one interior event. Hence, for the purposes of ease of discussion, we describe the PS knotting method for the case of a single interior event constraint.

PS Knotting Methods

To solve directly the Bolza problem posed in the preceding section, two basic discretizations are needed: one for the integral associated with the cost function and another for the dynamic constraints. We approximate the integral by a sum (using quadrature over the LGL or other PS points), whereas the derivative is approximated by a discrete differential operator. We focus on the Legendre PS method for the purpose of brevity; the extension to other PS methods is trivial.

Domain Transformation

In the Legendre PS approximation of the optimal control problem, the LGL node points lie in the computational interval $[-1, 1]$. The time coordinates $\tau^1 \in I^1 = [\tau_0, \tau_e]$ and $\tau^2 \in I^2 = [\tau_e, \tau_f]$ are related to $t \in [-1, 1]$ by the following affine transformations:

$$\tau^1 = [(\tau_e - \tau_0)t + (\tau_e + \tau_0)]/2, \qquad \tau^2 = [(\tau_f - \tau_e)t + (\tau_f + \tau_e)]/2$$

This results in the following reformulation of Eqs. (10), (16), and (13):

$$J[\boldsymbol{x}(\cdot), \boldsymbol{u}(\cdot), \tau_0, \tau_e, \tau_f, \boldsymbol{p}] = E\left(\boldsymbol{x}_0, \boldsymbol{x}_e^-, \boldsymbol{x}_e^+, \boldsymbol{x}_f; \boldsymbol{u}_e^-, \boldsymbol{u}_e^+; \tau_0, \tau_e, \tau_f; \boldsymbol{p}\right) + \frac{\tau_e - \tau_0}{2}\int_{-1}^{1} F^1(\boldsymbol{x}(t), \boldsymbol{u}(t), \tau(t), \boldsymbol{p})\,\mathrm{d}t + \frac{\tau_f - \tau_e}{2}\int_{-1}^{1} F^2(\boldsymbol{x}(t), \boldsymbol{u}(t), \tau(t), \boldsymbol{p})\,\mathrm{d}t \tag{27}$$

$$\boldsymbol{f}^{L^1} \le \boldsymbol{f}^1\left\{\left(\frac{2\dot{\boldsymbol{x}}(t)}{\tau_e - \tau_0}\right), \boldsymbol{x}(t), \boldsymbol{u}(t), \tau(t), \boldsymbol{p}\right\} \le \boldsymbol{f}^{U^1} \tag{28}$$

$$\boldsymbol{f}^{L^2} \le \boldsymbol{f}^2\left\{\left(\frac{2\dot{\boldsymbol{x}}(t)}{\tau_f - \tau_e}\right), \boldsymbol{x}(t), \boldsymbol{u}(t), \tau(t), \boldsymbol{p}\right\} \le \boldsymbol{f}^{U^2} \tag{29}$$

$$\boldsymbol{h}^{L^1} \le \boldsymbol{h}^1(\boldsymbol{u}(t), \boldsymbol{x}(t), \tau(t), \boldsymbol{p}) \le \boldsymbol{h}^{U^1}$$

$$\boldsymbol{h}^{L^2} \le \boldsymbol{h}^2(\boldsymbol{u}(t), \boldsymbol{x}(t), \tau(t), \boldsymbol{p}) \le \boldsymbol{h}^{U^2} \tag{30}$$

Strictly speaking, we must use different symbols for all of the mappings due to the transformation of the domain. However, we abuse notation here and retain these symbols for the purpose of brevity. Thus, in this context, one must view $\boldsymbol{x}(t)$ in the first segment, for example, as

$$\boldsymbol{x}(\tau(t)) = \boldsymbol{x}\{((\tau_e - \tau_0)t + (\tau_e + \tau_0))/2\} \in \mathbb{R}^{N_{x^1}}$$

for $\tau \in I^1$.

Problem Discretization

In the Legendre PS method, the LGL node points are closely related to the Legendre polynomials, which are orthogonal over the interval $[-1, 1]$ with respect to a unit weight function. Let $L_N(t)$ be the Legendre polynomial of degree N on the interval $[-1, 1]$. The LGL points[21] $t_l, l = 0, \ldots, N$, are given by

$$t_0 = -1, \qquad t_N = 1$$

and for $1 \le l \le N-1$, t_l are the zeros of $\dot{L}_N$, the derivative of the Legendre polynomial L_N. Our method may now be elaborated as follows: Let the integers $N^1 + 1$ and $N^2 + 1$ denote the number of LGL points on these subintervals and $t_l^i, i = 1, 2$, refer to the corresponding LGL points. The approximate solutions on these intervals I^i are denoted by $\boldsymbol{x}^{N^i}$ for states and $\boldsymbol{u}^{N^i}$ for controls, $i = 1, 2$,

$$\boldsymbol{x}(\tau) \approx \begin{cases} \boldsymbol{x}^{N^1}(\tau) & \tau \in I^1 \\ \boldsymbol{x}^{N^2}(\tau) & \tau \in I^2 \end{cases} \tag{31}$$

$$\boldsymbol{u}(\tau) \approx \begin{cases} \boldsymbol{u}^{N^1}(\tau) & \tau \in I^1 \\ \boldsymbol{u}^{N^2}(\tau) & \tau \in I^2 \end{cases} \tag{32}$$

where the superscript N^i is used for interval i, that is, whether or not the numerical value of N^1 is equal to N^2. The approximate states and controls are assumed to be a linear combination of Lagrange interpolating polynomials $\phi_l^i(t)$, so that we have

$$\boldsymbol{x}^{N^1}(\tau^1) = \sum_{l=0}^{N^1} \boldsymbol{x}_l^1 \phi_l^1(t), \qquad \boldsymbol{x}^{N^2}(\tau^2) = \sum_{l=0}^{N^2} \boldsymbol{x}_l^2 \phi_l^2(t) \tag{33}$$

$$\boldsymbol{u}^{N^1}(\tau^1) = \sum_{l=0}^{N^1} \boldsymbol{u}_l^1 \phi_l^1(t), \qquad \boldsymbol{u}^{N^2}(\tau^2) = \sum_{l=0}^{N^2} \boldsymbol{u}_l^2 \phi_l^2(t) \tag{34}$$

where $\tau^i = \tau(t^i)$, t is in the computational domain $[-1, 1]$ and $\tau_l^i = \tau^i(t_l^i)$, for $l = 0, \ldots, N^i, i = 1, 2$, are the shifted LGL points on interval I^i. Because $\phi_l^i(t)$ are interpolating polynomials, it follows that

$$\boldsymbol{x}_l^i = \boldsymbol{x}^{N^i}\left(\tau_l^i\right), \qquad \boldsymbol{u}_l^i = \boldsymbol{u}^{N^i}\left(\tau_l^i\right), \qquad \forall i, l$$

An expression for the Lagrange polynomials that interpolates the functions at the LGL points is given by[21]

$$\phi_l^i(t) = \frac{1}{N^i(N^i+1)L_{N^i}\left(t_l^i\right)} \frac{(t^2-1)\dot{L}_{N^i}(t)}{t - t_l^i} \tag{35}$$

for $l = 0, 1, \ldots, N^i, i = 1, 2$. The simple ideas described earlier are already quite revealing in how they apply to the limits described in Eq. (2). We assume that the operations of limits and approximations commute so that we can write

$$\boldsymbol{x}_e^- \approx \lim_{\tau \uparrow \tau_e} \boldsymbol{x}^{N^1}(\tau) = \boldsymbol{x}_{N_1}^1 \tag{36}$$

$$\boldsymbol{x}_e^+ \approx \lim_{\tau \downarrow \tau_e} \boldsymbol{x}^{N^2}(\tau) = \boldsymbol{x}_0^2 \tag{37}$$

Similarly, we get $\boldsymbol{u}_e^- \approx \boldsymbol{u}_{N_1}^1$ and $\boldsymbol{u}_e^+ \approx \boldsymbol{u}_0^2$. Thus, the right-hand sides of Eq. (2) are allowed to be distinct numbers in the approximation scheme. Because $\tau_{N^1}^1 = \tau_0^2 = \tau_e$, we can have $\lim_{\tau \uparrow \tau_e} \boldsymbol{x}(\tau) \ne \lim_{\tau \downarrow \tau_e} \boldsymbol{x}(\tau)$, when $N_{x^1} = N_{x^2}$.

To carry out the discretization of the problem, we impose the condition that the preceding approximations satisfy the differential inclusions at the LGL node points on the appropriate subinterval. To express the derivative $\dot{\boldsymbol{x}}^{N^i}[\tau^i(t)]$ in terms of $\boldsymbol{x}^{N^i}[\tau^i(t)]$ at the node points t_k^i, we differentiate Eq. (33) and evaluate the result at t_k^i to obtain a matrix multiplication of the following form for $i = 1, 2$:

$$\dot{\boldsymbol{x}}^{N^i}\left(\tau^i\left(t_k^i\right)\right) = \sum_{l=0}^{N^i} \boldsymbol{x}_l^i \dot{\phi}_l^i(t_k) = \sum_{l=0}^{N^i} D_{kl}^{N^i}\, \boldsymbol{x}_l^i \tag{38}$$

where $D_{kl}^{N^i} = \dot{\phi}_l^i(t_k)$ are entries of the $(N^i+1)\times(N^i+1)$ differentiation matrix $\boldsymbol{D}^{N^i}$

$$\boldsymbol{D}^{N^i} := \left[D_{kl}^{N^i}\right] := \begin{cases} \left[L_{N^i}\left(t_k^i\right)\Big/ L_{N^i}\left(t_l^i\right)\right]\cdot\left[1\Big/\left(t_k^i - t_l^i\right)\right] & k \neq l \\ -[N^i(N^i+1)/4] & k = l = 0 \\ N^i(N^i+1)/4 & k = l = N^i \\ 0 & \text{otherwise} \end{cases} \tag{39}$$

The discretization of the dynamic constraints can now be carried out in the following way:

$$\boldsymbol{f}^{L^1} \le \boldsymbol{f}^1\left([2/(\tau_e - \tau_0)]\dot{\boldsymbol{x}}_k^1, \boldsymbol{x}_k^1, \boldsymbol{u}_k^1, \tau_k^1, \boldsymbol{p}\right) \le \boldsymbol{f}^{U^1}, \qquad k = 0, \ldots, N^1$$

$$\boldsymbol{f}^{L^2} \le \boldsymbol{f}^2\left([2/(\tau_f - \tau_e)]\dot{\boldsymbol{x}}_k^2, \boldsymbol{x}_k^2, \boldsymbol{u}_k^2, \tau_k^2, \boldsymbol{p}\right) \le \boldsymbol{f}^{U^2}, \qquad k = 0, \ldots, N^2 \tag{40}$$

where the simplified, but somewhat abused, notation $\dot{\boldsymbol{x}}_k^i = \dot{\boldsymbol{x}}^{N^i}(\tau^i(t_k^i))$ is used. Next, with use of the Gauss–Lobatto integration rule (quadrature), the Bolza cost function in Eq. (27) is discretized, and the integrals are approximated by a finite sum,

$$J^N(\boldsymbol{X}, \boldsymbol{U}, \tau_0, \tau_e, \tau_f, \boldsymbol{p}) = E\left(\boldsymbol{x}_0^1, \boldsymbol{x}_{N_1}^1, \boldsymbol{x}_0^2, \boldsymbol{x}_{N_2}^2; \boldsymbol{u}_{N^1}^1, \boldsymbol{u}_0^2; \tau_0, \tau_e, \tau_f; \boldsymbol{p}\right)$$

$$+ \frac{\tau_e - \tau_0}{2} \sum_{k=0}^{N^1} F^1\left(\boldsymbol{x}_k^1, \boldsymbol{u}_k^1, \tau_k^1, \boldsymbol{p}\right) w_k^1$$

$$+ \frac{\tau_f - \tau_e}{2} \sum_{k=0}^{N^2} F^2\left(\boldsymbol{x}_k^2, \boldsymbol{u}_k^2, \tau_k^2, \boldsymbol{p}\right) w_k^2 \tag{41}$$

where

$$\boldsymbol{X} \equiv (\boldsymbol{X}^1, \boldsymbol{X}^2) = \left(\boldsymbol{x}_0^1, \boldsymbol{x}_1^1, \ldots, \boldsymbol{x}_{N^1}^1, \boldsymbol{x}_0^2, \boldsymbol{x}_1^2, \ldots, \boldsymbol{x}_{N^2}^2\right) \tag{42}$$

$$\boldsymbol{U} \equiv (\boldsymbol{U}^1, \boldsymbol{U}^2) = \left(\boldsymbol{u}_0^1, \boldsymbol{u}_1^1, \ldots, \boldsymbol{u}_{N^1}^1, \boldsymbol{u}_0^2, \boldsymbol{u}_1^2, \ldots, \boldsymbol{u}_{N^2}^2\right) \tag{43}$$

and w_k^i are the weights given by

$$w_k^i := [2/[N^i(N^i+1)]\left(1\Big/\left[L_{N^i}\left(t_k^i\right)\right]^2\right) \qquad i = 1, 2, \qquad k = 0, 1, \ldots, N \tag{44}$$

Thus, the event constraint is approximated by

$$\boldsymbol{e}^L \le \boldsymbol{e}\left(\boldsymbol{x}_0^1, \boldsymbol{x}_{N_1}^1, \boldsymbol{x}_0^2, \boldsymbol{x}_{N_2}^2; \boldsymbol{u}_{N^1}^1, \boldsymbol{u}_0^2; \tau_0, \tau_e, \tau_f; \boldsymbol{p}\right) \le \boldsymbol{e}^U \tag{45}$$

whereas the path constraints is discretized as

$$\boldsymbol{h}^{L^1} \le \boldsymbol{h}^1\left(\boldsymbol{x}_k^1, \boldsymbol{u}_k^1, \tau_k^1, \boldsymbol{p}\right) \le \boldsymbol{h}^{U^1}, \qquad k = 0, \ldots, N^1 \tag{46}$$

$$\boldsymbol{h}^{L^2} \le \boldsymbol{h}^2\left(\boldsymbol{x}_k^2, \boldsymbol{u}_k^2, \tau_k^2, \boldsymbol{p}\right) \le \boldsymbol{h}^{U^2}, \qquad k = 0, \ldots, N^2 \tag{47}$$

The optimal control problem is, hence, approximated to the mathematical programming problem of finding coefficients $\boldsymbol{X} = (\boldsymbol{X}^1, \boldsymbol{X}^2)$, $\boldsymbol{U} = (\boldsymbol{U}^1, \boldsymbol{U}^2)$, the parameter $\boldsymbol{p}$, and possibly the event times τ_0, τ_e, and τ_f that minimize the cost function (41) subject to the constraints given by Eq. (40) and Eqs. (45–47).

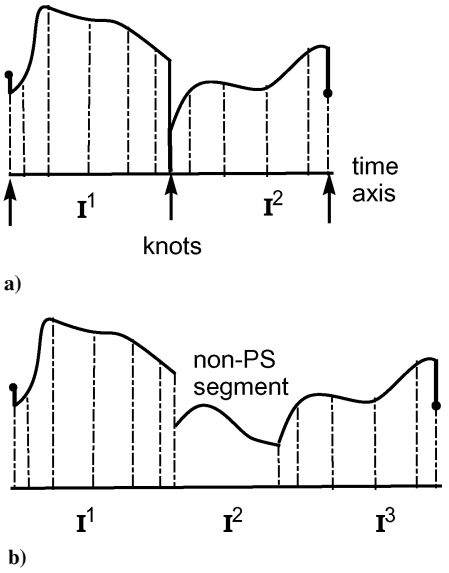

Fig. 2 Knots and nodes.

Knots and Nodes

PS discretization techniques for optimal control problems are distinctly different from the popular collocation methods including the new Runge–Kutta schemes (see Ref. 38). One major difference between PS methods and other methods is that PS methods are differentiation methods, whereas a majority of the other methods are integration methods. That is, PS methods discretize $\dot{x}$ directly and do not require f in $\dot{x} = f(x, u)$ to generate the approximation. In this sense, PS methods are similar to finite element in time methods,[39,40] but offer higher accuracy known as spectral accuracy.[41] In modern terminology,[42] PS methods rely on the discretization of the tangent bundle [roughly, the left-hand side of the differential equation, $\dot{x} = f(x, u)$] rather than an approximation of the vector field (the right-hand side). This is the reason why PS methods have only node points without extra collocation points. Because PS methods retain much of the differential-geometric properties as well as the functional-algebraic structures of the optimal control problem, they can be readily adapted to solve a rich variety of optimal control problems that include exploiting special properties. To illustrate the ramifications of the proposed method, note that $\tau_{N^1}^1 = \tau_0^2 = \tau_e = I^1 \cap I^2$; see the middle arrow in Fig. 2a. That is, we have a double node point at τ_e, which is the right Lobatto point of I^1 and the left Lobatto point of I^2. To distinguish these double-node points from the remainder of the single-node points (dashed vertical lines in Fig. 2), we use the word knots. Thus, knots facilitate an accurate characterization of multifunctions by allowing the value of the function at the knot to be multivalued. The knotting condition is Eq. (45) and is simply an approximation of the event condition [Eq. (4)], where the approximation stems from Eq. (36). Hence, the knotting condition transfers information at the knots across the two segments 1 and 2. We call such nodes hard knots, and they are intrinsic to the problem formulation. For example, the dropping of a stage, that is, mass discontinuity, in a multistage launch problem defines a hard knot. Because we may view $\boldsymbol{x}_0$ as $\lim_{\tau \to \tau_0} \boldsymbol{x}(\tau)$ and consequently define two distinct values of $\boldsymbol{x}_0$ as $\boldsymbol{x}_0^-$ and $\boldsymbol{x}_0^+$, it is apparent that we can regard the initial conditions (event) as part of the general framework of knots as illustrated by the left arrow in Fig. 2a. The same argument applies to the final-time conditions as well, as illustrated by the right arrow in Fig. 2a. Problems with possible end point knots arise in problems such as aeroassisted maneuvers, where initial and final delta-V are part of the problem formulation.

Whether or not a problem has hard event conditions, we can use the concept of knots for various other purposes by defining a false event of state continuity at an arbitrary point τ_e,

$$\boldsymbol{x}_e^- \equiv \boldsymbol{x}_e^+$$

leading to a linear knotting condition

$$\boldsymbol{x}_{N_1}^1 - \boldsymbol{x}_0^2 = \boldsymbol{0} \tag{48}$$

We call such knots soft knots. Soft knots have many advantages because Eq. (48) can be imposed at any point where the state is continuous; see Eq. (3) and the discussion immediately following it. Because continuity of the control is not imposed at the soft knot, the control can take any admissible value at no cost; hence, the knot facilitates an accurate representation of a control switch, if it exists. In the examples to follow, soft knots are used to capture accurately nonsmooth behavior such as corners in states and switches in the control. We use the terms fixed (hard/soft) knot if τ_e is fixed and free (hard/soft) knot if τ_e is free. Clearly, fixed/free hard knots are based on problem formulation and fixed/free soft knots are designer knots.

Non-PS Segments

The concept of knots can also be used to incorporate non-PS segments[5,13] as demonstrated in Ref. 5. To illustrate this notion, consider a three-segment solution over the three subintervals, $I^1 = [\tau_0, \tau_1]$, $I^2 = [\tau_1, \tau_2]$, and $I^3 = [\tau_2, \tau_f]$ (Fig. 2b). Suppose that the trajectory over I^2 is a predefined segment, for example, a coast arc over an interplanetary trajectory or a straight-line segment in an obstacle-free zone in the path of a mobile robot. In such cases, it is clear that it is not necessary to discretize I^2. We now define our event times as τ_1 and τ_2 and let

$$\boldsymbol{x}_e^- = \lim_{\tau \uparrow \tau_1} \boldsymbol{x}(\tau), \qquad \boldsymbol{x}_e^+ = \lim_{\tau \downarrow \tau_2} \boldsymbol{x}(\tau) \tag{49}$$

[Compare Eq. (2).] Thus, the map $\boldsymbol{x}_e^- \mapsto \boldsymbol{x}_e^+$ can be quite arbitrary and becomes part of the event condition, and only the intervals I^1 and I^3 are discretized. Because the map $\boldsymbol{x}_e^- \mapsto \boldsymbol{x}_e^+$ does not even have to be an analytic expression, it is obvious that a direct-shooting segment is implied in this formulation. In the same vein, any other discretization can be performed between two knots. Thus, for example, we may discretize I^2 by a standard collocation method and use the knotting conditions to connect non-PS segments. One reason for this flexibility is that the PS knotting method is inherently parallel in the sense that much of the computations that occur between two adjacent knots can be done independently of those over other knots.

Using Knots for Efficient Scaling

The concept of knots can also be used for effectively scaling a numerical problem, an important practical step in balancing equations for computational efficiency. This notion can best be explained by a simple example. Consider the problem of optimizing a low-thrust interplanetary trajectory. If heliocentric-based canonical units are uniformly used over multiple planetary encounters, the equations of motion will be ill balanced near the planetary encounters and can cause major difficulties in convergence. On the other hand, if the units are switched near the planetary encounters (for example, planet-centric canonical units) for balancing the equations, the differently scaled variables exhibit an apparent discontinuity although the actual variables may be continuous. Because it is the scaled variable that is eventually being solved for, the apparent discontinuity can be easily captured by the concept of knots, where the knotting condition is simply a continuity of the real variable modeled as an event condition in the form of a discontinuity of the scaled variable. An example of such an application of knots is discussed in Ref. 37.

Software

The PS knotting method is implemented in the object-oriented reusable software package DIDO.[13] No knowledge of PS methods is necessary to use DIDO because all of the concepts delineated earlier are automated. As an outcome of the ideas described in this paper, significant generality in problem formulation is allowed. For example, the cost function may be a function of the derivative $\dot{\boldsymbol{x}}$, as in a neoclassical approach.[32] DIDO exploits the problem solving environment of MATLAB® and uses the suite of mathematical programming solvers available through TOMLAB.[43] Although the default solver in DIDO is SNOPT,[44] a sequential quadratic programming (SQP) method,[45] identical results (within the numerical accuracy of the plots) were also obtained by the use of a new concept of the filter SQP method[46] implemented in filterSQP[47] and an interior point method[48] implemented in KNITRO.[49] In all cases, the sparsity pattern of the PS method is explicitly exploited in various DIDO constructs, as well as through the TOMLAB solver suite. A typical sparsity pattern is shown in Fig. 3. In the following section, we illustrate some of the concepts. Applications to significantly more complex problems may be found elsewhere, for example, Refs. 5, 7, 9, 37, and 50.

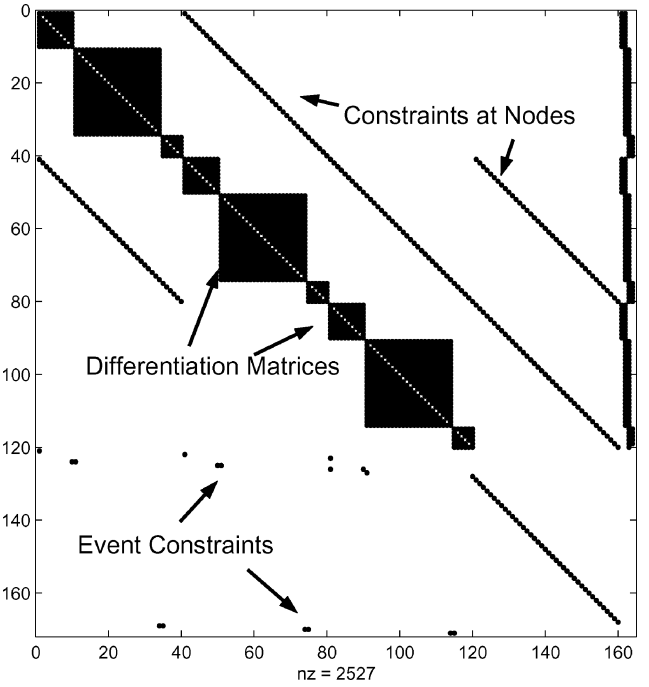

Fig. 3 Typical sparsity pattern for the PS knotting method: white space represents the zero elements.

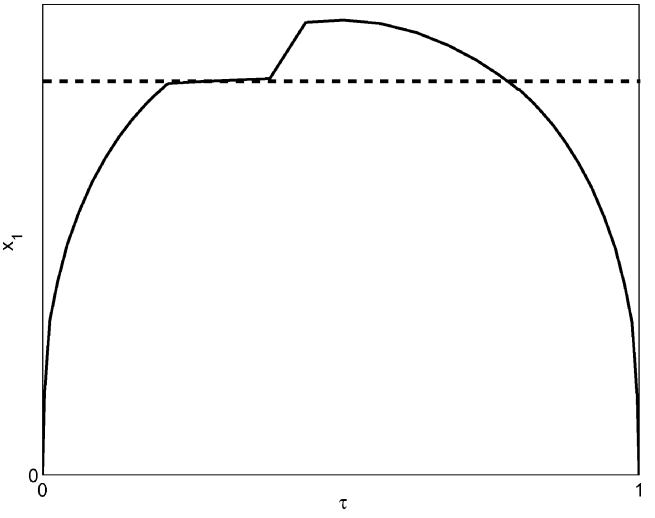

Fig. 4 Schematic of the nonsmooth problem of Queen Dido in the Badlands.

Example 1: Queen Dido and the Badlands

This is the famous nonsmooth problem of Clarke[18] based on the classic smooth isoperimetric problem: Queen Dido is given a rope of fixed length L and she has to enclose a region of maximum area between two points, $(0, 0)$ and $(1, 0)$, along a straight shore represented by $x_1 = 0$ in the $\tau - x_1$ plane (Fig. 4). Unlike the classical problem,

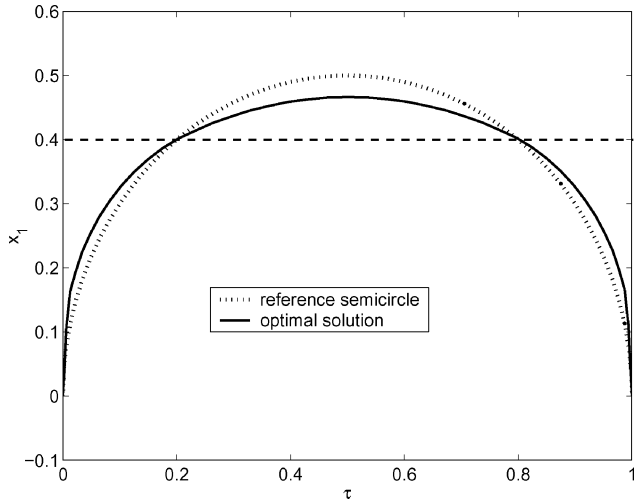

Fig. 5 Solution to Queen Dido's nonsmooth problem.

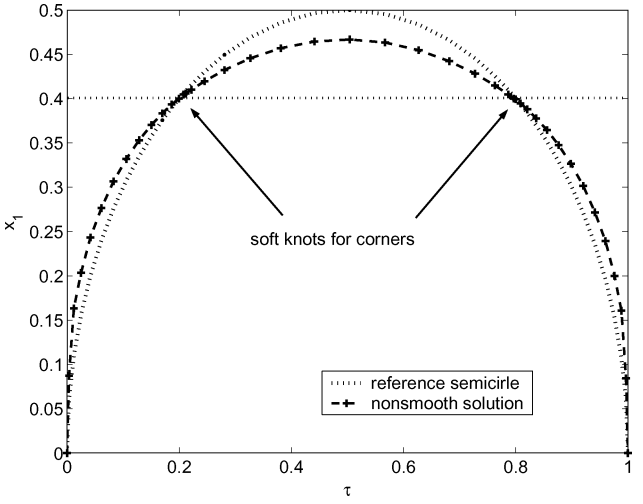

Fig. 6 Solution to Queen Dido's nonsmooth problem with soft knots. Note that the soft knots migrate toward the corners.

a more realistic problem formulation requires that we account for value of the terrain toward the shore being greater than that away from it. In Clarke's nonsmooth formulation, the terrain for $x_1 \geq \alpha$ (dashed line in Fig. 4) is worth half as much as the terrain $x_1 \leq \alpha$, where α is some given positive number. When the Mayer formulation suggested by Clarke is used, the problem can be posed as

$$\text{maximize} \quad J = x_2(\tau_f) \tag{50}$$

$$\text{subject to} \quad \dot{x}_1 = u \tag{51}$$

$$\dot{x}_2 = \begin{cases} x_1 & \text{if} \quad x_1 \leq \alpha \\ (x_1 + \alpha)/2 & \text{if} \quad x_1 \geq \alpha \end{cases} \tag{52}$$

$$\dot{x}_3 = \sqrt{1 + u^2} \tag{53}$$

$$x_1(0) = x_2(0) = x_3(0) = 0 \tag{54}$$

$$x_1(\tau_f) = 0 \tag{55}$$

$$x_3(\tau_f) \leq \pi \tag{56}$$

Note that in the problem formulation what is also unknown is the number and sequence of phases in addition to a dwell time, if any, over the line $x_1 = \alpha$ (Fig. 4). In other words, a nonsmooth direct method should be able to determine such segments automatically without introducing combinatorial issues related to phases. That the PS method can indeed do this is demonstrated in Fig. 5 where it conforms to Clarke's analytical solution of piecewise circular subarcs. The solution for the smooth problem, that is, constant value of the terrain, is simply a semicircle and is shown in Fig. 5 for comparison. For the numerical simulation, we chose $\alpha = 0.4$ and $N = 45$ somewhat arbitrarily. The solution from the PS method agrees with Clarke's theoretical result and reveals that there is no subarc over the line $x_1 = \alpha$; compare Fig. 5 with Fig. 4. Furthermore, the optimal solution consists of three subarcs with corners at the property line where the value halves. From Fig. 5, it is apparent that one corner occurs in the subinterval [0, 0.5] and another one over [0.5, 1]. To capture the corners more accurately, we introduce two free soft knots. To demonstrate that a more accurate result is possible by the introduction of these soft knots, we now split the 45 node points into three equal divisions of 15 points. This result is shown in Fig. 6. From Fig. 6, it is apparent that free soft knots can be used to locate corners accurately in the optimal solution.

Example 2: Vertical Ascent of a Two-Stage Rocket

This problem is chosen to illustrate the concept of hard knots and the mixing of hard and soft knots. Solutions to significantly more complex launch-vehicle trajectory optimization problems, including guidance via real-time optimization, are discussed in Refs. 7 and 50.

The problem is to maximize the final altitude during the vertical ascent of a two-stage rocket. This is a time-free problem of minimizing

$$J = -r(\tau_f) \tag{57}$$

subject to the dynamic constraints

$$\dot{r} = v \tag{58}$$

$$\dot{v} = -\mu/r^2 + T/m \tag{59}$$

$$\dot{m} = -T/v_e \tag{60}$$

where the state variables r, v, and m are radius, speed, and mass, respectively. The constant parameters in the problem are the normalized gravitational constant $\mu = 1$ and the normalized exhaust velocity of the rocket, $v_e = 0.5$.

The rocket has two stages. The maximum thrust force on the rocket is different over each stage and for stage 1 is given by

$$0 \leq T \leq T^1_{\max} = 1.167 \tag{61}$$

and for stage 2 is

$$0 \leq T \leq T^2_{\max} = 0.875 \tag{62}$$

The propellent in the first stage is 0.2 units and the drop mass is 0.1 units. Hence, at some unknown time τ_e, we have a jump discontinuity in the mass given by $m^-(\tau_e) - m^+(\tau_e) = 0.1$. We formulate the switching condition as

$$r^-(\tau_e) - r^+(\tau_e) = 0 \tag{63}$$

$$v^-(\tau_e) - v^+(\tau_e) = 0 \tag{64}$$

$$m^+(\tau_e) = 0.7 \tag{65}$$

$$0.1 \leq m^-(\tau_e) - m^+(\tau_e) \leq 0.3 \tag{66}$$

where the last inequality is simply written to allow the PS method to drop fuel with the structure. Although it is obvious that fuel should not be dropped with the structure mass for maximizing the altitude, this condition is allowed as a self-check on the optimal solution. In any case, this interior event serves as the hard knot for this problem. The remainder of the hard knots are given by the boundary conditions

$$r(0) = 1.0, \qquad v(0) = 0, \qquad m(0) = 1.0, \qquad m(\tau_f) = 0.5 \tag{67}$$

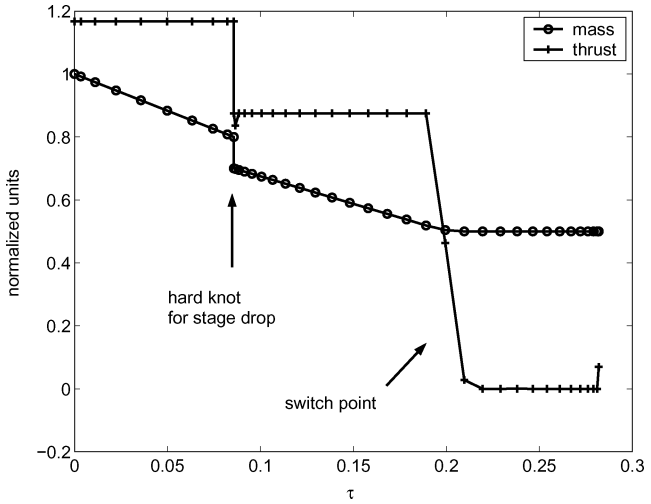

Fig. 7 Vertical ascent of the two-stage rocket problem.

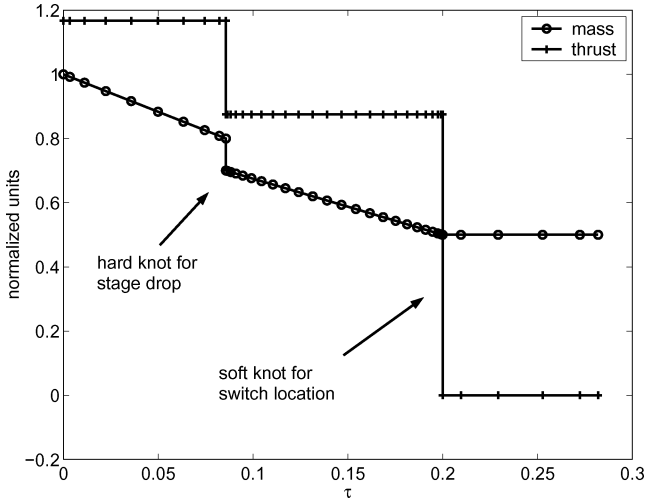

Fig. 8 Mixing of hard and soft knots for the two-stage rocket problem.

A solution to this problem using the hard-knot option of the PS method is shown in Fig. 7. Note the set-valued representation at the hard knot (first stage) and the Gibbs phenomenon near the switch point during the second stage. This solution was obtained using 40 node (LGL) points with 10 points for the first stage and 30 points for the second stage. What is immediately apparent from Fig. 7 is the accurate representation of switches in both the thrust and mass variables, particularly at the point of stage drop. At the point of stage drop, mass and thrust are represented as multifunctions. Note also that only the structure mass of the vehicle is dropped despite the fact that we allowed the code to drop some or all of the first-stage fuel along with the dry mass. That the final result does do as expected provides a self-check on the validity of the solution. The solution also reveals that there is a switch in the thrust magnitude (from its maximum value of the second stage to zero) at some point $\tau_e \in [0.15, 0.25]$. That we should have a zero thrust arc for the terminal subarc is intuitively obvious and provides a second self-check on the validity of the solution.

To better represent the switch point over the flight of the second stage, it is apparent that we can use a free soft knot over this portion of flight. The solution with the introduction of a soft knot (but with the same total number of nodes) is shown in Fig. 8. Note the set-valued representation at all knots. It is apparent that the absence of a soft knot smears the discontinuity in the thrust, an artifact of the Gibbs phenomenon, whereas the knotting method works well in capturing switches. Note also that the knots cluster at precisely the switching points; indeed, this is a natural artifact of the PS knotting method. This point is further illustrated in Fig. 9, where the plots of the radius and velocity are shown. It is apparent from Fig. 9 that the nondifferentiability in the velocity variable is captured very well by the PS knotting method.

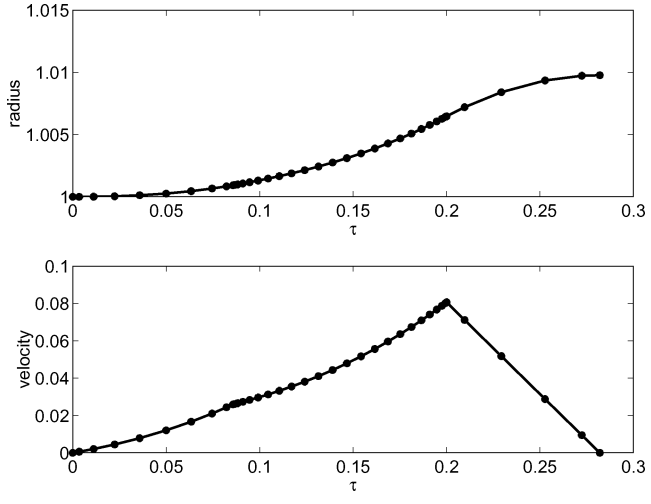

Fig. 9 Radius and velocity for the two-stage rocket problem: representation of nonsmooth functions with mixed hard and soft knots.

Conclusions

The concept of PS knots was formally introduced although its implementation in the software package DIDO has been rather widely used in the literature before the current exposition. PS knotting methods offer great flexibility in solving smooth, nonsmooth, and hybrid optimal control problems. Smooth optimal control problems, that is, problems with smooth data, may generate nonsmooth trajectories, particularly due to mixed state-control constraints. General computational methods for solving nonsmooth problems are lacking in the literature despite their wide applicability to solving practical engineering problems. This paper has demonstrated that PS knots can be effectively used to address difficulties encountered in solving such practical problems. Our proposed method can efficiently handle a vast number of other situations arising in real-world optimal control problems such as rapid changes in dynamics, state-dependent control constraints, switching conditions, and generalized event conditions associated with hybrid optimal control problems.

Acknowledgments

We gratefully acknowledge the financial support provided in part by (listed in alphabetical order) Charles Stark Draper Laboratory, Inc. (Draper), Jet Propulsion Laboratory (JPL), Naval Postgraduate School, and the Secretary of the U.S. Air Force. In particular, we thank Timothy J. Brand (Draper) and Steven E. Matousek (JPL) for challenging us with complex problems and facilitating the astronautical applications of pseudospectral knotting methods. We also thank our graduate student, James B. Ross, for providing some of the initial input files to DIDO that helped us obtain quick solutions to the two-stage ascent problem. We reserve our special gratitude for Ronald J. Proulx for providing valuable suggestions and insights.

References

[1]Fahroo, F., and Ross, I. M., "Costate Estimation by a Legendre Pseudospectral Method," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 2, 2001, pp. 270–277.

[2]Stanton, S., Proulx, R. J., and D'Souza, C. N., "Optimal Orbit Transfer Using a Legendre Pseudospectral Method," American Astronautical Society, AAS Paper 03-574, Aug. 2003.

[3]Strizzi, J., Ross, I. M., and Fahroo, F., "Towards Real-Time Computation of Optimal Controls for Nonlinear Systems," AIAA Paper 2002-4945, Aug. 2002.

[4]Melton, R. G., "Comparison of Direct Optimization Methods Applied to Solar Sail Problems," AIAA Paper 2002-4728, Aug. 2002.

[5]Stevens, R., and Ross, I. M., "Preliminary Design of Earth-Mars Cyclers Using Solar Sails," *Journal of Spacecraft and Rockets* (to be published); also American Astronautical Society, Paper AAS 03-244, Feb. 2003.

[6]Lu, P., Sun, H., and Tsai, B., "Closed-Loop Endoatmospheric Ascent Guidance," *Journal of Guidance, Control and Dynamics*, Vol. 26, No. 2, 2003, pp. 283–294.

[7]Rea, J., "Launch Vehicle Trajectoy Optimization Using a Legendre Pseudospectral Method," AIAA Paper 2003-5640, Aug. 2003.

[8]Josselyn, S., and Ross, I. M., "A Rapid Verification Method for the Trajectory Optimization of Reentry Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 3, 2003, pp. 505–508.

[9]Fahroo, F., Doman, D., and Ngo, A., "Modeling Issues in Footprint Generation for Reusable Launch Vehicles," *Proceedings of the IEEE Aerospace Conference*, Vol. 6, IEEE Press, New York, 2003, pp. 2791–2799.

[10]Yan, H., Fahroo, F., and Ross, I. M., "Accuracy and Optimality of Direct Transcription Methods," *American Astronautical Society*, Paper AAS 00-205, Jan. 2000.

[11]Proulx, R. J., and Ross, I. M., "Time-Optimal Reorientation of Asymmetric Rigid Bodies," American Astronautical Society/AIAA Astrodynamics Specialist Conf., AAS 01-384, July–Aug. 2001.

[12]Williams, P., Blanksby, C., and Trivailo, P., "Receding Horizon Control of Tether System Using Quasilinearization and Chebyshev Pseudospectral Approximations," American Astronautical Society, Paper AAS 03-535, Aug. 2003.

[13]Ross, I. M., and Fahroo, F., "User's Manual for DIDO 2002: A MATLAB Application Package for Dynamic Optimization," Dept. of Aeronautics and Astronautics, NPS Technical Rept. AA-02-002, Naval Postgraduate School, Monterey, CA, June 2002.

[14]Elnagar, J., Kazemi, M. A., 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.

[15]Fahroo, F., and Ross, I. M., "A Second Look at Approximating Differential Inclusions," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 1, 2001, pp. 131–133.

[16]Ross, I. M., and Fahroo, F., "User's Manual for DIDO 2001: A MATLAB Application Package for Dynamic Optimization," Dept. of Aeronautics and Astronautics, NPS Technical Rept. AA-01-003, Naval Postgraduate School, Monterey, CA, June 2001.

[17]Ross, I. M., and Fahroo, F., "A Direct Method for Solving Nonsmooth Optimal Control Problems," *Proceedings of the 2002 World Congress of the International Federation on Automatic Control*, IFAC, Barcelona, Spain, 2002.

[18]Clarke, F. H., *Optimization and Nonsmooth Analysis*, Wiley-Interscience, New York, 1983; reprinted as *Classics in Applied Mathematics*, Vol. 5, Society for Industrial and Applied Mathematics, Philadelphia, 1990.

[19]Betts, J. T.,"Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 193–207.

[20]Betts, J. T., *Practical Methods for Optimal Control Using Nonlinear Programming*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2001.

[21]Canuto, C., Hussaini, M. Y., Quarteroni, A., and Zang, T.A., *Spectral Methods in Fluid Dynamics*, Springer-Verlag, New York, 1988.

[22]Gottlieb, D., Hussaini, M. Y., and Orszag, S. A., "Theory and Applications of Spectral Methods," *Spectral Methods for PDE's*, edited by R. G. Voigt, D. Gottlieb, and M. Y. Hussaini, Society for Industrial and Applied Mathematics, Philadelphia, 1984.

[23]Banks, H. T., and Fahroo, F., "Legendre–Tau Approximations for LQR Feedback Control of Acoustic Pressure Fields," Summary, *Journal of Mathematical Systems, Estimation, and Control*, Vol. 5, No. 2, 1995, pp. 271–274.

[24]Elnagar, G., and Kazemi, M. A., "Pseudospectral Chebyshev Optimal Control of Constrained Nonlinear Dynamical Systems," *Computational Optimization and Applications*, Vol. 11, 1998, pp. 195–217.

[25]Fahroo, F., and Ross, I. M., "Direct Trajectory Optimization by a Chebyshev Pseudospectral Method," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 1, 2002, pp. 160–166.

[26]Ross, I. M., Rea, J., and Fahroo, F., "Exploiting Higher-Order Derivatives in Computational Optimal Control," *Proceedings of the 10th Mediterranean Conference on Control and Automation*, Mediterranean Control Assoc., 2002.

[27]Ross, I. M., and Fahroo, F., "Pseudospectral Methods for Optimal Motion Planning of Differentially Flat Systems," *IEEE Transactions on Automatic Control* (to be published).

[28]Ross, I. M., and Fahroo, F., "Legendre Pseudospectral Approximations of Optimal Control Problems," *Lecture Notes in Control and Information Sciences*, Vol. 295, Springer-Verlag, Berlin, 2003.

[29]Fornberg, B., *A Practical Guide to Pseudospectral Methods*, Cambridge Univ. Press, New York, 1998.

[30]Fahroo, F., and Ross, I. M., "A Spectral Patching Method for Direct Trajectory Optimization," *Journal of the Astronautical Sciences*, Vol. 48, No. 2/3, 2000, pp. 269–286.

[31]Sussmann, H. J., "A Maximum Principle for Hybrid Optimal Control Problems," *Proceedings of the 38th IEEE Conference on Decision and Control*, IEEE Press, Piscataway, NJ, 1999.

[32]Vinter, R., *Optimal Control*, Birkhäuser, Boston, MA, 2000.

[33]Sussmann, H. J., "Some Recent Results On The Maximum Principle of Optimal Control Theory," *Systems and Control in the 21st Century*, edited by C. I. Byrnes, B. N. Datta, D. S. Gilliam, and C. F. Martin, Birkhäuser, Boston, 1997, pp. 351–372.

[34]Stewart, D. E., "Rigid-Body Dynamics with Friction and Impact," *SIAM Review*, Vol. 42, No. 1, 2000, pp. 3–39.

[35]Dontchev, A., and Lempio, F., "Difference Methods for Differential Inclusions: A Survey," *SIAM Review*, Vol. 34, No. 2, 1992, pp. 263–294.

[36]Ross, I. M., and Fahroo, F., "A Perspective on Methods for Trajectory Optimization," AIAA Paper 2002-4727, Aug. 2002.

[37]Josselyn, S. B., "Optimization of Low-Thrust Trajectories with Terminal Aerocapture," Aeronautical and Astronautical Engineering Degree Thesis, Naval Postgraduate School, Monterey, CA, Dec. 2002.

[38]Hager, W. W., "Runge–Kutta Methods in Optimal Control and the Transformed Adjoint System," *Numerische Mathematik*, Vol. 87, 2000, pp. 247–282.

[39]Hodges, D., and Bless, R., "Weak Hamiltonian Finite-Element Method for Optimal Control Problems," *Journal of Guidance, Control, and Dynamics*, Vol. 14, 1991, pp. 148–156.

[40]Warner, M., and Bless, R., " Solving Optimal Control Problems Using *hp*-Version Finite Elements in Time," *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 1, 2000, pp. 86–94.

[41]Trefethen, L. N., *Spectral Methods in MATLAB*, Society for Industrial and Applied Mathematics, Philadelphia, 2000.

[42]Sussmann, H. J., "Geometry and Optimal Control," *Mathematical Control Theory*, edited by J. Baillieul and J. C. Willems, Springer-Verlag, New York, 1998, pp. 140–198.

[43]Holmström, K., Göran, A. O., and Edvall, M. M., "User's Guide for TOMLAB 4.0.6," TOMLAB Optimization AB, Vasteras Sweden, Aug. 2003.

[44]Gill, P. E., Murray, W., and Saunders, M. A., "User's Guide for SNOPT Version 6: A FORTRAN Package for Large-scale Nonlinear Programming," Univ. of California, San Diego, La Jolla, CA, Dec. 2002.

[45]Gill, P. E., Murray, W., and Saunders, M.A., "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Journal on Optimization*, Vol. 12, No. 4, 2002, pp. 979–1006.

[46]Fletcher, R., and Leyffer, S., "Nonlinear Programming Without a Penalty Function," Univ. of Dundee Numerical Analysis Rept. NA/171, Dundee, Scotland, U.K., Aug. 2000.

[47]Fletcher, R., and Leyffer, S., "User's Manual for filterSQP," Univ. of Dundee Numerical Analysis Report, Dundee, Scotland, U.K., March 1999.

[48]Byrd, R. H., Hribar, M. H., and Nocedal, J., "An Interior Point Algorithm for Large-Scale Nonlinear Programming," *SIAM Journal on Optimization*, Vol. 9, No. 4, pp. 877–900.

[49]Waltz, R. A., and Nocedal, J., "KNITRO User's Manual: Version 3.0," Northwestern Univ., Technical Rept. OTC 2003/5, Evanston, IL, April 2003.

[50]Ross, I. M., D'Souza, C. N., Fahroo, F., and Ross, J. B., "A Fast Approach to Multi-Stage Launch Vehicle Trajectory Optimization," AIAA Paper 2003-5639, Aug. 2003.