JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 32, No. 4, July–August 2009

Engineering Notes

ENGINEERING NOTES are short manuscripts describing new developments or important results of a preliminary nature. These Notes should not exceed 2500 words (where a figure or table counts as 200 words). Following informal review by the Editors, they may be published within a few months of the date of receipt. Style requirements are the same as for regular contributions (see inside back cover).

Hermite–Legendre–Gauss–Lobatto Direct Transcription in Trajectory Optimization

Paul Williams*
Delft University of Technology,
2600 AA Delft, The Netherlands

DOI: 10.2514/1.42731

Introduction

DIRECT methods have been widely applied for solving trajectory optimization problems [1–5]. Among the popular methods are the Hermite–Simpson method [6] and the Legendre pseudospectral (PS) method [7,8]. There has been some considerable interest in developing theory related to the Legendre PS method due to its high accuracy, although the Hermite–Simpson method continues to be applied to large-scale practical problems [9]. The main characteristic of the Hermite–Simpson method is the combination of reasonable accuracy with a highly sparse constraint Jacobian and Hessian matrix [10]. The PS method offers impressive accuracy (spectral accuracy) for smooth problems, but the constraint Jacobians are much denser than other methods. The sparsity of the constraint Jacobians can be increased in the PS method by using knots [11].

In addition to the Hermite–Simpson method, additional high-order methods have been proposed by Herman and Conway [2]. These methods are attractive from the point of view of accuracy, but in the framework proposed by Herman and Conway, they require detailed derivation when extended to arbitrary higher orders. For instance, in [2], the form of the constraints was derived via the symbolic manipulation software MAPLE.

The purpose of this Note is to provide an alternative framework for arbitrary higher-order methods suitable for implementation on digital computers and in a reusable form. The optimal control problem is approximated by a discrete nonlinear programming problem (NLP) by expanding the state trajectories using local Hermite interpolating polynomials. For high accuracy, the collocation points are selected from the family of Gauss–Lobatto points. This also allows the integral performance index to be approximated via Gauss–Lobatto quadrature rules. For optimal control problems of the Bolza form, the natural choice of quadrature points are the Legendre–Gauss–Lobatto (LGL) points, because they are derived on the basis of a unity weight function, giving the highest accuracy for polynomial integrands. The generalization of the approach is referred to as the Hermite–Legendre–Gauss–Lobatto (HLGL) approach throughout the remainder of this Note.

Received 14 December 2008; revision received 16 March 2009; accepted for publication 22 March 2009.

*Applied Researcher, Faculty of Aerospace Engineering; tethered.systems@gmail.com.

Mathematical Framework

Optimal Control Problem

To facilitate the discussion, a rather general optimal control problem may be considered. Consider the problem of determining the state-control pair $\{\boldsymbol{x}(\cdot), \boldsymbol{u}(\cdot)\}$ to minimize the cost function

$$\mathcal{J} = \mathcal{E}[\boldsymbol{x}(t_0), \boldsymbol{x}(t_f), t_0, t_f] + \int_{t_0}^{t_f} \mathcal{L}[\boldsymbol{x}(t), \boldsymbol{u}(t), t]\mathrm{d}t \tag{1}$$

subject to the nonlinear state equations

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \tag{2}$$

the endpoint conditions

$$\boldsymbol{e}_L^0 \le \boldsymbol{e}[\boldsymbol{x}(t_0), t_0] \le \boldsymbol{e}_U^0 \tag{3}$$

$$\boldsymbol{e}_L^f \le \boldsymbol{e}[\boldsymbol{x}(t_f), t_f] \le \boldsymbol{e}_U^f \tag{4}$$

path constraints

$$\boldsymbol{p}_L \le \boldsymbol{p}[\boldsymbol{x}(t), \boldsymbol{u}(t), t] \le \boldsymbol{p}_U \tag{5}$$

and box constraints

$$\boldsymbol{x}_L \le \boldsymbol{x}(t) \le \boldsymbol{x}_U \qquad \boldsymbol{u}_L \le \boldsymbol{u}(t) \le \boldsymbol{u}_U \tag{6}$$

where $\boldsymbol{x} \in \mathbb{R}^{n_x}$ are the state variables, $\boldsymbol{u} \in \mathbb{R}^{n_u}$ are the control inputs, $t \in \mathbb{R}$ is the time, $\mathcal{E}$: $\mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is the Mayer cost function, $\mathcal{L}$: $\mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \to \mathbb{R}$ is the integrand of the Bolza cost function, $\boldsymbol{e}_L^0 \in \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_0}$ and $\boldsymbol{e}_U^0 \in \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_0}$ are the lower and upper bounds on the initial point conditions, $\boldsymbol{e}_L^f \in \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_f}$ and $\boldsymbol{e}_U^f \in \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_f}$ are the lower and upper bounds on the final point conditions, and $\boldsymbol{p}_L \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \to \mathbb{R}^{n_p}$ and $\boldsymbol{p}_U \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \to \mathbb{R}^{n_p}$ are the lower and upper bounds on the path constraints. This optimal control problem is discretized and then solved using nonlinear programming, as detailed in the following subsection.

Numerical Discretization

For boundary value problems, the Gauss–Lobatto points are a natural choice for nodes because they combine high accuracy of interpolation with highly accurate quadrature approximations. Two popular choices for Gauss–Lobatto points have been used in the literature. The Legendre–Gauss–Lobatto [3,8] and Chebyshev-Gauss–Lobatto (CGL) [4,5] points are based on the roots of the derivatives of Legendre and Chebyshev polynomials, respectively, which are two instances of the more general Jacobi polynomials. The LGL points minimize the L^2-norm of the approximation error, whereas the CGL points minimize the max-norm of the approximation error. When N nodes are used, the LGL quadrature rule is exact for polynomials of the order $2N + 1$, whereas when the CGL points are used, the corresponding Curtis–Clenshaw quadrature rule is only exact for polynomials of order N. The approach used in this

Note can be extended to arbitrary Gauss–Lobatto grids and quadrature rules [12], with the LGL nodes being the most practical.

The method used to derive the residual in [6] can be extended to arbitrary-order polynomials and implemented numerically without recourse to deriving the analytical expressions for the residuals as in [2]. In the derivation presented here, it is necessary to consider only odd orders of polynomials n for the state approximations. This is because the number of constraints per interval is $(n-1)/2$. For example, in the Hermite–Simpson method ($n=3$), the number of constraints per interval is one, whereas for the fifth-order method derived by Herman and Conway [2] ($n=5$), the number of constraints per interval is two.

Briefly, the basis of the approach for formulating the optimal control problem as a nonlinear programming problem is the discretization of the state equations and performance index using a polynomial expansion. The state equations are approximated by formulating a set of residual equations which are driven to zero by the NLP. These residual equations, sometimes called defects, are the subject of this Note.

The procedure for formulating the constraints is to approximate the state by first dividing the entire time interval for the problem into m subintervals. The state in the ith interval is approximated by the nth degree Hermite interpolating polynomial:

$$\boldsymbol{x}(\tau) \approx \boldsymbol{a}_0 + \boldsymbol{a}_1\tau + \boldsymbol{a}_2\tau^2 + \boldsymbol{a}_3\tau^3 + \cdots + \boldsymbol{a}_n\tau^n, \qquad \tau \in [-1, 1] \tag{7}$$

Note that strictly speaking, subscripts for the ith subinterval should be employed. However, to keep things simple, these have been dropped with the understanding that Eq. (7) is applied independently in each of the m subintervals. To ensure a good combination of interpolation accuracy and integration accuracy, the nodes and collocation points within each interval are defined as the LGL points ξ_j ($j=1,\ldots,n$), which are the zeros of the derivative of the $(n-1)$ th-degree Legendre polynomial $L_{n-1}(\xi)$ in the interval $[-1,1]$ including the endpoints. In this Note, a distinction between nodes and collocation points is made as follows: the nodes are used to form the interpolating polynomial, whereas the collocation points are used to formulate the residual equations for the NLP. Note that in some methods, such as the Legendre pseudospectral method [3], there is no distinction between nodes and collocation points (i.e., the collocation points are the nodes and vice versa). The LGL points give maximum accuracy for quadrature approximations as well as avoiding the Runge phenomenon during interpolation. A graphical representation of the nodes and collocation points used in this work is given in Fig. 1.

The state is approximated at the node points, defined according to

$$\tau_j \triangleq \xi_{2j-1}, \qquad j = 1, \ldots, (n+1)/2 \tag{8}$$

The coefficients of the Hermite interpolating $\boldsymbol{a}_i$ ($i=1,\ldots,n$) are determined by using the values of the states and state derivatives at the points τ_j:

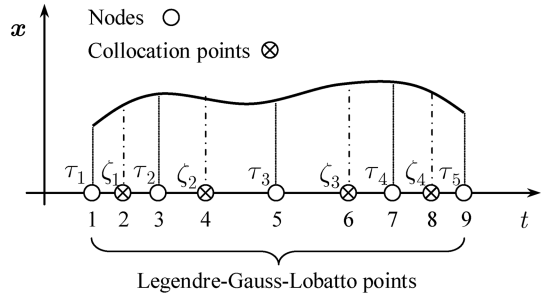

Fig. 1 Definition of nodes and collocation points for Hermite–Legendre–Gauss–Lobatto discretization.

$$\begin{bmatrix} 1 & -1 & 1 & \cdots & \tau_1^{n-1} & \tau_1^n \\ 1 & \tau_2^1 & \tau_2^2 & \cdots & \tau_2^{n-1} & \tau_2^n \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \tau_{\frac{n+1}{2}}^1 & \tau_{\frac{n+1}{2}}^2 & \cdots & \tau_{\frac{n+1}{2}}^{n-1} & \tau_{\frac{n+1}{2}}^n \\ 0 & 1 & 2\tau_1^1 & \cdots & (n-1)\tau_1^{n-2} & n\tau_1^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 2\tau_{\frac{n+1}{2}}^1 & \cdots & (n-1)\tau_{\frac{n+1}{2}}^{n-2} & n\tau_{\frac{n+1}{2}}^{n-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{a}_0 \\ \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \\ \vdots \\ \vdots \\ \boldsymbol{a}_{n-1} \\ \boldsymbol{a}_n \end{bmatrix}$$

$$= \begin{bmatrix} \boldsymbol{x}(\tau_1) \\ \boldsymbol{x}(\tau_2) \\ \vdots \\ \boldsymbol{x}(\tau_{\frac{n+1}{2}}) \\ \frac{h_i}{2}\boldsymbol{f}(\tau_1) \\ \vdots \\ \frac{h_i}{2}\boldsymbol{f}(\tau_{\frac{n+1}{2}}) \end{bmatrix} \tag{9}$$

where it is assumed that the state equations are of the form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \tag{10}$$

and the interval length is defined by $h_j \triangleq t_{j+1} - t_j$.

Note that the left-hand side of Eq. (9) depends only on the number and location of the node points. Therefore, given a vector of states, one can solve Eq. (9) for the coefficients of the Hermite interpolating polynomial in the ith subinterval in terms of the values of the states and vector field at the same nodes. Denote the solution to Eq. (9) as $\boldsymbol{a} = [\boldsymbol{a}_0, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_n]^T$, then $\boldsymbol{a} = [\boldsymbol{A}]^{-1}\boldsymbol{b}$, where $\boldsymbol{A}$ is the matrix on the left-hand side of Eq. (9), and $\boldsymbol{b}$ is the concatenated vector of states and scaled state derivatives on the right-hand side of Eq. (9).

The values of the states at the constraint collocation points, defined according to

$$\zeta_j = \xi_{2j}, \qquad j = 1, \ldots, (n-1)/2 \tag{11}$$

may be found from the interpolation formula

$$\boldsymbol{x}(\zeta_j) = (\boldsymbol{v}_j^\top [\boldsymbol{A}]^{-1})\boldsymbol{b} \triangleq \phi_j^T \boldsymbol{b}, \qquad j = 1, \ldots, (n-1)/2 \tag{12}$$

where

$$\boldsymbol{v}_j = [1 \quad \zeta_j \quad \zeta_j^2 \quad \cdots \quad \zeta_j^n], \qquad j = 1, \ldots, (n-1)/2 \tag{13}$$

The constraints are enforced by evaluating the state equations using $\boldsymbol{x}(\zeta_j)$ in Eq. (12) and equating the result with the derivative of the Hermite interpolating polynomial at the collocation points:

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{x}(\zeta_j)}{\mathrm{d}\tau} &= \boldsymbol{x}' = \boldsymbol{a}_1 + 2\boldsymbol{a}_2\zeta_j + 3\boldsymbol{a}_3\zeta_j^2 + \cdots + n\boldsymbol{a}_n\zeta_j^{n-1} \\ &= (\boldsymbol{v}_{d_j}^T [\boldsymbol{A}]^{-1})\boldsymbol{b} \triangleq \phi_j'^T \boldsymbol{b} \end{aligned} \tag{14}$$

where

$$\boldsymbol{v}_{d_j} = [0 \quad 1 \quad 2\zeta_j \quad \cdots \quad n\zeta_j^{n-1}], \qquad j = 1, \ldots, (n-1)/2 \tag{15}$$

By collecting the coefficients ϕ_j and ϕ_j' into matrices

$$\Phi = [\phi_1, \phi_2, \ldots, \phi_{(n-1)/2}], \qquad \Phi' = [\phi_1', \phi_2', \ldots, \phi_{(n-1)/2}'] \tag{16}$$

the states and state derivatives can be obtained using matrix-vector operations. Thus, we obtain a system of $(n-1)/2$ constraints per interval:

$$\boldsymbol{\Delta}_i = \begin{bmatrix} \boldsymbol{x}'(\zeta_1) - \frac{h_i}{2}\boldsymbol{f}(\boldsymbol{x}(\zeta_1), \boldsymbol{u}(\zeta_1), t(\zeta_1)) \\ \boldsymbol{x}'(\zeta_2) - \frac{h_i}{2}\boldsymbol{f}(\boldsymbol{x}(\zeta_2), \boldsymbol{u}(\zeta_2), t(\zeta_2)) \\ \vdots \\ \boldsymbol{x}'(\zeta_{\frac{n-1}{2}}) - \frac{h_i}{2}\boldsymbol{f}(\boldsymbol{x}(\zeta_{\frac{n-1}{2}}), \boldsymbol{u}(\zeta_{\frac{n-1}{2}}), t(\zeta_{\frac{n-1}{2}})) \end{bmatrix} = \Phi^T \boldsymbol{b}$$
$$- \frac{h_i}{2}\boldsymbol{f}(\Phi^T \boldsymbol{b}, \boldsymbol{u}(\zeta), t(\zeta)) = \boldsymbol{0} \tag{17}$$

In this form, because the matrix of coefficients are constants, independent of the states or time, analytic Jacobians for the NLP can be derived in a straightforward manner using the chain rule for the implicit differentiation in Eq. (17).

This procedure is equivalent to the Hermite–Simpson method for $n = 3$ and equivalent to the fifth-order method derived by Herman and Conway [2] for $n = 5$, although there is no need to write the constraints explicitly in this approach. Also note that the matrices in Eq. (16) only need to be calculated once numerically, before solving the NLP or stored in a file. Therefore, in terms of implementation, the number of arithmetic operations (pure multiplication) in evaluating the constraints will always be faster than the evaluation of the corresponding explicit constraints in [2], which contain square roots and divides. However, if the explicit constraints in [2] are precomputed, then the approach presented here will be equivalent in the number of arithmetic operations.

The cost function is approximated by a Gauss–Lobatto quadrature rule as

$$\mathcal{J} \cong \mathcal{M}(\boldsymbol{x}_n^0, \boldsymbol{x}_n^N, t_0, t_N) + \sum_{i=1}^{N} \frac{h_i}{2} \sum_{j=1}^{n} w_j \mathcal{L}(\boldsymbol{x}_j^{(i)}, \boldsymbol{u}_j^{(i)}, t_j^{(i)}) \tag{18}$$

where

$$w_j = \frac{2}{(n-1)n} \frac{1}{[L_{n-1}(\xi_j)]^2}, \qquad j = 1, \ldots, n \tag{19}$$

Some rather extensive work has been undertaken in previous studies to determine the accuracy of the transcribed dual-optimal-control problem [8,13,14]. It is known that some methods give an exact relationship between the discretized indirect problem and the Karush–Kuhn–Tucker (KKT) version of the direct problem (such as the Euler method, some Runge-Kutta methods [13], and the Legendre pseudospectral method). It is known that the Hermite–Simpson method gives a lower order of accuracy in the discrete covectors than it does for the state equations [14]. Nevertheless, the KKT multipliers are still accurate and the dual problem converges to the infinite dimensional problem as the number of subintervals becomes small [15].

The discrete approximations to the costates for the HLGL discretization may be obtained at the collocation points from the corresponding Lagrange multipliers using the following mapping:

$$\boldsymbol{\lambda}(\zeta_k) \approx \tilde{\boldsymbol{\lambda}}_k / w_{2k}, \qquad k = 1, \ldots, (n-1)/2 \tag{20}$$

where $\tilde{\boldsymbol{\lambda}}_k$ are the Lagrange multipliers associated with the constraint equations (17). This relationship is *not exact*, because the interpolating polynomial formed using the coefficients in Eq. (20) is of lower order than the state approximation.

The HLGL approach allows the trading of polynomial degree n with the number of intervals. For smooth problems, better accuracy can be achieved by increasing n, whereas for nonsmooth problems, increasing the number of subintervals can lead to better accuracy. This offers possibilities for mesh refinement for Hermite-based approaches beyond those already in use [10].

Numerical Example

The transcription of the optimal control problem is implemented via a MATLAB interface, and the NLP is solved using the sparse sequential quadratic programming software SNOPT [16], originally coded in FORTRAN, but called from MATLAB via a mex-file interface. Note that the implementation has not been optimized for speed. For example, the states at all node points are used as decision variables, even though the states at the constraint collocation points may be obtained from the states at the nodes using Eq. (12). In the interests of comparison, the method of [2] has been implemented using identical constructs to the approach used in this Note. Jacobians are provided via finite differencing.

The example of shuttle reentry from [17] is used. The problem is to find a trajectory to minimize the final latitude of the shuttle, subject to initial and terminal point constraints, as well as a heating rate constraint. The equations of motion for the vehicle over a spherical, rotating Earth are given by

$$\dot{r} = v \sin\gamma \tag{21}$$

$$\dot{\theta} = v \cos\gamma \cos\psi / (r \cos\phi) \tag{22}$$

$$\dot{\phi} = v \cos\gamma \sin\psi / r \tag{23}$$

$$\dot{v} = -D/m - g \sin\gamma + \Omega^2 r \cos\phi (\sin\gamma \cos\phi - \cos\gamma \sin\phi \sin\psi) \tag{24}$$

$$\dot{\psi} = L \sin\sigma / (mv \cos\gamma) - v \cos\gamma \cos\psi \tan\phi / r - r\Omega^2 \sin\phi \cos\phi \cos\psi / (v \cos\gamma) + 2\Omega(\tan\gamma \cos\phi \sin\psi - \sin\phi) \tag{25}$$

$$\dot{\gamma} = (L \cos\sigma/m - g \cos\gamma)/v + v \cos\gamma / r + r\Omega^2 \cos\phi (\cos\gamma \cos\phi + \sin\gamma \sin\phi \sin\psi)/v + 2\Omega \cos\phi \cos\psi \tag{26}$$

where $g = \mu_e / r^2$ is the gravitational acceleration at radius r, θ is the longitude, ϕ is the latitude, v is the velocity, ψ is the heading angle, γ is the flight-path angle, Ω is the Earth's angular velocity, m is the

Table 1 Comparison of numerical solutions for shuttle reentry problem

Method	Mean CPU time	Std dev CPU time	Cost function
Explicit fifth order (10 intervals)	59.5	11.5	−0.501944
Numerical fifth order (10 intervals)	49.3	7.7	−0.501944
Seventh order, 8 intervals	84.4	30.5	−0.501641
Ninth order, 6 intervals	71.0	16.4	−0.501821
Explicit fifth order (20 intervals)	284.1	26.9	−0.501781
Numerical fifth order (20 intervals)	258.7	27.5	−0.501781
Seventh order, 15 intervals	455.1	142.9	−0.501795
Ninth order, 11 intervals	408.1	126.1	−0.501814
Explicit fifth order (30 intervals)	577.3	80.6	−0.501786
Numerical fifth order (30 intervals)	519.6	64.3	−0.501786
Seventh order, 22 intervals	713.2	103.6	−0.501785
Ninth order, 17 intervals	786.6	85.6	−0.501783

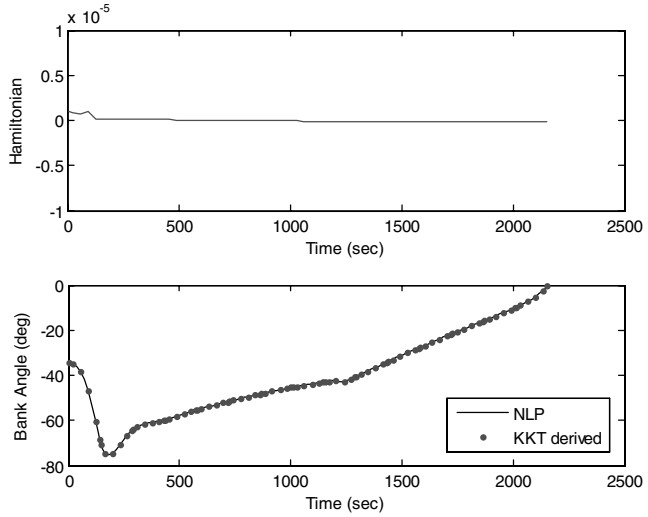

Fig. 2 Example results from reentry problem: a) Hamiltonian derived from discrete costates ($H = 0$), b) bank angle obtained from NLP and that obtained from Pontryagin's maximum principle and discrete costates.

vehicle mass, $L(\alpha)$ is the lift, $D(\alpha)$ is the drag, α is the angle of attack, and σ is the bank angle. The aerodynamic and heating models can be found in [17]. Optimal trajectories are determined for initial conditions $h = r - R_e = 260,000$ ft, $v = 24,061$ ft/s, $\phi = 0$, $\psi = 0$, $\theta = 0$, and $\gamma = -1.064$ deg and for the terminal conditions $h = 80,000$ ft, $v = 2500$ ft/s, and $\gamma = -5$ deg.

The calculated high-order solution using a polynomial degree of 13 with 15 intervals gives a terminal latitude of -0.501783. A comparison of the implementation of the numerical constraint with the explicit constraint for a fifth-order ($n = 5$) solution, together with higher-order solutions, is given in Table 1. Comparisons are made in terms of cost function and computation times. The higher-order solutions are chosen to consist of roughly the same number of decision variables as the fifth-order solutions. In each case, the initial guesses are zero controls, with randomized state trajectories. The same guesses are provided to each method for a particular set of random seeds to prevent skewing the results. A total of 50 samples were used to produce the results in this work. The results show, as expected, that precomputing the weighting coefficients results in faster solution times than evaluating the closed-form coefficients. As noted previously, the difference in solution time will become zero if the closed-form coefficients are precomputed. The results show that using higher-order polynomials results in improvements in accuracy at the expense of increases in runtime, due to the denser Jacobians. Figure 2a shows the Hamiltonian derived by means of the discrete costates obtained using Eq. (20). For the time-free problem, the necessary condition for optimality is for the Hamiltonian to be zero. This necessary condition is accurate to the order of the precision of the NLP for the constraint equations (i.e., 10^{-6}). Figure 2b shows the optimal bank angle, together with the bank angle obtained via the discrete costates. This shows that the costates are accurate enough to provide a means for partially verifying optimality. The ability to freely change the polynomial order for particular sections of the trajectory, instead of simply increasing the number of intervals, makes the formulation viable for enhancing mesh refinement algorithms.

Conclusions

The well-known direct transcription methods belonging to a set of Gauss–Lobatto quadrature rules have been extended to arbitrary order within a simple numerical framework. The method uses Hermite interpolating polynomials to approximate the state equations, which employ the Legendre–Gauss–Lobatto points as interpolation and collocation points. This unifies the computation of the constraint equations and integral cost function in the method. Essentially, this approach is a direct extension to higher orders of the Hermite–Simpson and a fifth-order method proposed in earlier work. This offers possibilities for improving mesh refinement procedures by choosing different polynomial orders within particular regions, in combination with increasing the number of intervals.

References

[1] Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 193–207. doi:10.2514/2.4231

[2] Herman, A. L., and Conway, B. A., "Direct Optimization Using Collocation Based on High-Order Gauss–Lobatto Quadrature Rules," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 3, 1996, pp. 592–599. doi:10.2514/3.21662

[3] Elnagar, G., 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. doi:10.1109/9.467672

[4] Elnagar, G. M., and Kazemi, M. A., "Pseudospectral Chebyshev Optimal Control of Constrained Nonlinear Dynamical Systems," *Computational Optimization and Applications*, Vol. 11, No. 2, 1998, pp. 195–217. doi:10.1023/A:1018694111831

[5] 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. doi:10.2514/2.4862

[6] Hargraves, C. R., and Paris, S. W., "Direct Trajectory Optimization Using Nonlinear Programming and Collocation," *Journal of Guidance, Control, and Dynamics*, Vol. 10, No. 4, 1987, pp. 338–342. doi:10.2514/3.20223

[7] 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. doi:10.2514/2.4709

[8] 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, pp. 327–342.

[9] Betts, J. T., and Erb, S. O., "Optimal Low Thrust Trajectories to the Moon," *SIAM Journal on Applied Dynamical Systems*, Vol. 2, No. 2, pp. 144–170. doi:10.1137/S1111111102409080

[10] Betts, J. T., *Practical Methods for Optimal Control Using Nonlinear Programming*, Advances in Control and Design Series, Society for Industrial and Applied Mathematics, Philadelphia, 2001.

[11] Ross, I. M., and Fahroo, F., "Pseudospectral Knotting Methods for Solving Nonsmooth Optimal Control Problems," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 3, 2004, pp. 397–405. doi:10.2514/1.3426

[12] Williams, P., "Nonlinear Control and Applications of Tethered Space Systems," Ph.D. Dissertation, RMIT University, Melbourne, Australia, 2004.

[13] Hager, W. W., "Runge-Kutta Method in Optimal Control and the Transformed Adjoint System," *Numerische Mathematik*, Vol. 87, No. 2, 2000, pp. 247–282. doi:10.1007/s002110000178

[14] Enright, P. J., and Conway, B. A., "Discrete Approximations to Optimal Trajectories Using Direct Transcription and Nonlinear Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 4, 1992, pp. 994–1002. doi:10.2514/3.20934

[15] von Stryk, O., and Burlisch, R., "Direct and Indirect Methods for Trajectory Optimization," *Annals of Operations Research*, Vol. 37, No. 1, 1992, pp. 357–373. doi:10.1007/BF02071065

[16] 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. doi:10.1137/S1052623499350013

[17] Josselyn, S., and Ross, I. M., "Rapid Verification Method for the Trajectory Optimization of Reentry Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 3, 2003, pp. 505–508. doi:10.2514/2.5074