[6]Palacián, J., "Teoría del Satélite Artificial: Armónicos Teserales y su Relegación Mediante Simplificaciones Algebraicas," Ph.D. Dissertation, Dept. Matemática Aplicada, Univ. de Zaragoza, Spain, May 1992.

[7]Segerman, A. M., and Coffey, S. L., "An Analytical Theory for Tesseral Gravitational Harmonics," *Celestial Mechanics and Dynamical Astronomy*, Vol. 76, No. 3, 2000, pp. 139–156.

[8]Abad, A., and San-Juan, J. F., "PSPC: A Poisson Series Processor Coded in C," *Dynamics and Astrometry of Natural and Artificial Celestial Bodies*, edited by K. Kurzyńska, F. Barlier, P. K. Seidelmann, and I. Wytrzyszczak, Astronomical Obeservatory of A. Mickiewicz University, Poznań, Poland, 1994, pp. 383–388.

[9]Abad, A., Elipe, A., Palacián, J., and San Juan, J. F., "ATESAT: A Symbolic Processor for Artificial Satellite Theory," *Mathematics and Computers in Simulation*, Vol. 45, Nos. 5–6, 1998, pp. 497–510.

[10]Scheeres, D. J., "The Effect of $C_{2,2}$ on Orbit Energy and Angular Momentum," *Celestial Mechanics and Dynamical Astronomy*, Vol. 73, Nos. 1–4, 1999, pp. 339–348.

[11]Hu, W., and Scheeres, D. J., "Spacecraft Motion about Slowly Rotating Asteroids," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 4, 2002, pp. 466–475.

[12]Whittaker, E. T., *A Treatise on the Analytical Dynamics of Particles and Rigid Bodies*, Cambridge Univ. Press, Cambridge, UK, 1904, P. 349.

[13]Hill, G. W., "Motion of a System of Material Points Under the Action of Gravitation," *Astronamical Journal*, Vol. 27, Nos. 22, 23, 1913, pp. 171–182.

[14]Miller, J. K., Konopliv, A. S., Antreasian, P. G., Bordi, J. J., Chesley, S., Helfrich, C. E., Owen, W. M., Scheeres, D. J., Wang, T. C., Williams, B. G., and Yeomans, D. K., "Determination of Shape, Gravity and Rotational State of Asteroid 433 Eros," *Icarus*, Vol. 155, No. 1, 2002, pp. 3–17.

[15]Broucke, R., "Numerical Integration of Periodic Orbits in the Main Problem of Artificial Satellite Theory," *Celestial Mechanics and Dynamic Astronomy*, Vol. 58, No. 2, 1994, pp. 99–123.

Jacobi Pseudospectral Method for Solving Optimal Control Problems

Paul Williams*

RMIT University, Melbourne, Victoria, 3001 Australia

Introduction

THE solution of optimal control problems via direct methods has become popular in recent years and has been used to solve a wide variety of problems successfully.[1] The most common direct methods can be grouped into 1) local and 2) global methods. Local methods have been termed direct collocation[2] or direct transcription.[3] In local methods, a series of node points of arbitrary spacing are defined at which both the state and control vectors are collocated. The state equations are enforced as equality constraints at internal collocation points between the nodes by implicit integration techniques such as Simpson's rule (see Ref. 4) or by Gauss–Lobatto quadrature rules (see Ref. 2). In other words, the state equations are enforced locally. Global or pseudospectral methods use globally orthogonal interpolating polynomials based on the Gauss–Lobatto points for Legendre (see Refs. 5 and 6) or Chebyshev (see Refs. 7 and 8) polynomials to approximate the state and control variables. The state equations are enforced by differentiating the approximating polynomial at the corresponding Gauss–Lobatto points (which are the zeros of the derivative of the interpolating polynomial), rather than through numerical integration. Global methods are expected to be more accurate than local methods because the discrete adjoint multipliers retain the same order of accuracy as the state equations, which is not true for certain classes of local methods [such as the Hermite–Simpson (see Ref. 3) and particular Runge–Kutta discretizations (see Ref. 9)]. Ross and Fahroo[10] discuss these ideas in detail.

The use of pseudospectral methods for solving optimal control problems has been restricted in the literature to either Legendre or Chebyshev methods. In this Note, the pseudospectral method is generalized to consider collocation based on the roots of the derivatives of general Jacobi polynomials. The Legendre and Chebyshev nodes can be obtained as particular cases of the more general formulation. Comparisons between different nodes can be obtained by simply changing the parameters in the Jacobi polynomial. Appropriate selection of the Jacobi parameters may form an important part in determining real-time solutions to nonlinear optimal control problems.

Numerical Method

Statement of the Problem

Consider the problem of minimizing the performance index

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

where $t \in R$, $\boldsymbol{x} \in R^n$, and $\boldsymbol{u} \in R^m$ are subject to the dynamic constraints

$$f[\ddot{\boldsymbol{x}}(t), \dot{\boldsymbol{x}}(t), \boldsymbol{x}(t), \boldsymbol{u}(t), t] = \boldsymbol{0}, \qquad t \in [t_0, t_f] \tag{2}$$

and boundary conditions

$$\boldsymbol{\psi}_0[\dot{\boldsymbol{x}}(t_0), \boldsymbol{x}(t_0), t_0] = \boldsymbol{0} \tag{3}$$

$$\boldsymbol{\psi}_f[\dot{\boldsymbol{x}}(t_f), \boldsymbol{x}(t_f), t_f] = \boldsymbol{0} \tag{4}$$

where $\boldsymbol{\psi}_0 \in R^p$ and $\boldsymbol{\psi}_f \in R^q$, with $p \leq n$ and $q \leq n$, and the state and control constraints

$$\boldsymbol{g}[\dot{\boldsymbol{x}}(t), \boldsymbol{x}(t), \boldsymbol{u}(t)] \leq \boldsymbol{0}, \qquad \boldsymbol{g} \in R^r \tag{5}$$

Discretization of the State Equations

The Jacobi pseudospectral method for solving optimal control problems, like the Legendre and Chebyshev methods, is based on expanding the state and control trajectories using Lagrange interpolating polynomials. An arbitrary selection of node points can lead to poor interpolation characteristics such as the Runge phenomenon, and so the node points in pseudospectral methods are chosen as the Gauss–Lobatto points. In this Note, the nodal points are obtained as the extrema of the Nth order Jacobi polynomial $P_N^{(\alpha,\beta)}$, where $\alpha > -1$ and $\beta > -1$ are parameters that determine the characteristic of the polynomial. Jacobi polynomials are orthogonal over the interval $(-1, 1)$ with respect to the weight function $w(\tau) = (1-\tau)^\alpha(1+\tau)^\beta$ and are the eigenfunctions of the Sturm–Liouville problem (see Ref. 11)

$$(1-\tau^2)P_N^{\prime\prime(\alpha,\beta)} + [(\beta-\alpha) - (\alpha+\beta+2)\tau]P_N^{\prime(\alpha,\beta)} + N(N+\alpha+\beta+1)P_N^{(\alpha,\beta)} = 0 \tag{6}$$

The Legendre and Chebyshev polynomials belong to the class of Jacobi polynomials and may be obtained by setting $\alpha = \beta = 0$ for Legendre and $\alpha = \beta = -0.5$ for Chebyshev.

Received 23 July 2003; revision received 13 October 2003; accepted for publication 20 October 2003. Copyright © 2003 by Paul Williams. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0731-5090/04 $10.00 in correspondence with the CCC.

*Ph.D. Candidate, Department of Aerospace Engineering. Student Member AIAA.

In pseudospectral methods, the nodal points $\tau_k, k=0,\ldots,N$, lie in the interval $\tau \in [-1,1]$ and include the endpoints, $(\tau_0=-1, \tau_N=1)$. These points are referred to as the Jacobi–Gauss–Lobatto (JGL) points, where τ_k, $1 \le k \le N-1$, are the zeros of $P_N^{\prime(\alpha,\beta)}$ and $()' = \mathrm{d}()/\mathrm{d}\tau$. The JGL nodes, τ_j, $j=0,\ldots,N$, may be found by determining the eigenvalues of a modified Jacobi matrix, as outlined by Gautschi.[12] Because the nodes for the interpolating polynomials lie in the domain $\tau \in [-1,1]$, a linear transformation is required to map the computational domain to the physical domain $t \in [t_0, t_f]$,

$$t = (t_f - t_0)\tau/2 + (t_f + t_0)/2 \tag{7}$$

The state and control variables are approximated by Nth-degree polynomials

$$\boldsymbol{x}_N(\tau) = \sum_{j=0}^{N} \hat{\boldsymbol{x}}_j \phi_j(\tau) \tag{8}$$

$$\boldsymbol{u}_N(\tau) = \sum_{j=0}^{N} \hat{\boldsymbol{u}}_j \phi_j(\tau) \tag{9}$$

where $\hat{\boldsymbol{x}}_j, \hat{\boldsymbol{u}}_j$, $j=0,\ldots,N$, are the coefficients of the interpolating polynomial. If it is desired that the coefficients $\hat{\boldsymbol{x}}_j = \boldsymbol{x}_N(\tau_j)$ and $\hat{\boldsymbol{u}}_j = \boldsymbol{u}_N(\tau_j)$, then it is clear that $\phi_j(\tau)$ are the Lagrange interpolating polynomials (derivation in Appendix A)

$$\phi_j(\tau) = \prod_{\substack{k=0\\k\ne j}}^{N}\left(\frac{\tau-\tau_k}{\tau_j-\tau_k}\right) = \frac{(\tau^2-1)}{(\tau-\tau_j)}\frac{c_j P_N^{\prime(\alpha,\beta)}(\tau)}{N(N+\alpha+\beta+1)P_N^{(\alpha,\beta)}(\tau_j)}$$

$$j=0,\ldots,N \tag{10}$$

where

$$c_j = \begin{cases} \beta+1, & j=0\\ 1, & 1\le j\le N-1 \\ \alpha+1, & j=N \end{cases} \tag{11}$$

To approximate the system dynamics, it is necessary to find expressions of the first and second derivatives of the approximating polynomials at the JGL nodes. The derivatives of the approximating functions can be obtained by directly differentiating Eq. (8) to obtain the Jacobi differentiation matrices

$$\boldsymbol{d}_k^{(1)} = \boldsymbol{x}_N'(\tau_k) = \sum_{j=0}^{N}\hat{\boldsymbol{x}}_j\phi_j'(\tau_k) = \sum_{j=0}^{N}\mathcal{D}_{kj}^{(1)}\hat{\boldsymbol{x}}_j = [\boldsymbol{D}\boldsymbol{x}_N]_k \tag{12}$$

$$\boldsymbol{d}_k^{(2)} = \boldsymbol{x}_N''(\tau_k) = \sum_{j=0}^{N}\hat{\boldsymbol{x}}_j\phi_j''(\tau_k) = \sum_{j=0}^{N}\mathcal{D}_{kj}^{(2)}\hat{\boldsymbol{x}}_j = [\boldsymbol{D}_2\boldsymbol{x}_N]_k \tag{13}$$

where the coefficients $\mathcal{D}_{kj}^{(1)}$ and $\mathcal{D}_{kj}^{(2)}$ are entries of $(N+1)\times(N+1)$ differentation matrices, $\boldsymbol{D}$ and $\boldsymbol{D}_2$, respectively, and subscript k indicates the kth element. Note that the state derivatives at any point depend on the values of the states at every node point, which must be contrasted with the way the state equations are approximated in other collocation methods (such as the Hermite–Simpson method). For the JGL points, the differentiation matrix may be written as (derivation in Appendix B)

$$\mathcal{D}_{kj} = \begin{cases}
\dfrac{\alpha - N(N+\alpha+\beta+1)}{2(\beta+2)} & k=j=0\\[2mm]
\dfrac{N(N+\alpha+\beta+1)-\beta}{2(\alpha+2)} & k=j=N\\[2mm]
\dfrac{P_N^{(\alpha,\beta)}(\tau_k)}{P_N^{(\alpha,\beta)}(\tau_j)}\dfrac{1}{(\tau_k-\tau_j)} & 1\le k\ne j\le N-1\\[2mm]
\dfrac{(\alpha+\beta)\tau_k+\alpha-\beta}{2\left(1-\tau_k^2\right)} & 1\le k=j\le N-1\\[2mm]
-\dfrac{P_N^{(\alpha,\beta)}(\tau_0)}{P_N^{(\alpha,\beta)}(\tau_j)}\dfrac{1}{(\beta+1)(1+\tau_j)} & k=0,\quad 1\le j\le N-1\\[2mm]
\dfrac{P_N^{(\alpha,\beta)}(\tau_N)}{P_N^{(\alpha,\beta)}(\tau_j)}\dfrac{1}{(\alpha+1)(1-\tau_j)} & k=N,\quad 1\le j\le N-1\\[2mm]
\dfrac{P_N^{(\alpha,\beta)}(\tau_j)}{P_N^{(\alpha,\beta)}(\tau_0)}\dfrac{(\beta+1)}{(1+\tau_j)} & j=0,\quad 1\le k\le N-1\\[2mm]
-\dfrac{P_N^{(\alpha,\beta)}(\tau_j)}{P_N^{(\alpha,\beta)}(\tau_N)}\dfrac{(\alpha+1)}{(1-\tau_j)} & j=N,\quad 1\le k\le N-1\\[2mm]
-\dfrac{(\alpha+1)}{2(\beta+1)}\dfrac{P_N^{(\alpha,\beta)}(\tau_0)}{P_N^{(\alpha,\beta)}(\tau_N)} & k=0,\quad j=N\\[2mm]
\dfrac{(\beta+1)}{2(\alpha+1)}\dfrac{P_N^{(\alpha,\beta)}(\tau_N)}{P_N^{(\alpha,\beta)}(\tau_0)} & k=N,\quad j=0
\end{cases} \tag{14}$$

The second derivative matrix $\boldsymbol{D}_2$ is obtained as $\boldsymbol{D}_2 = (\boldsymbol{D})^2$.

The derivatives of the interpolating polynomials given in Eqs. (12) and (13) are derivatives with respect to the parameter τ and not the independent variable in the original problem, t. Derivatives with respect to t are obtained as follows

$$\frac{\mathrm{d}\boldsymbol{x}_N}{\mathrm{d}t} = \frac{\mathrm{d}\boldsymbol{x}_N}{\mathrm{d}\tau}\frac{\mathrm{d}\tau}{\mathrm{d}t} = \frac{2}{(t_f-t_0)}\frac{\mathrm{d}\boldsymbol{x}_N}{\mathrm{d}\tau} = \frac{2}{(t_f-t_0)}\boldsymbol{D}\boldsymbol{x}_N \tag{15}$$

The state equations (2) may be discretized as equality constraints as follows:

$$\boldsymbol{f}\left\{\left[4\big/(t_f-t_0)^2\right]\boldsymbol{d}_k^{(2)}, [2/(t_f-t_0)]\boldsymbol{d}_k^{(2)}, \boldsymbol{x}_k, \boldsymbol{u}_k, t_k\right\} = \boldsymbol{0}$$

$$k=0,\ldots,N \tag{16}$$

The discretization of the higher-order derivatives and its advantages in reducing the number of optimization variables has been discussed by Ross et al.[13] and will not be elaborated on here.

Discretization of the Performance Index

The performance index may be approximated using a modified spectral averaging method originally outlined by Trefethen.[14] Consider the integration of the function $f(t)$ over the domain $[t_0, t_f]$,

$$\mathcal{I} = \int_{t_0}^{t_f} f(t)\,\mathrm{d}t = \frac{(t_f-t_0)}{2}\int_{-1}^{1} f(\tau)\,\mathrm{d}\tau \tag{17}$$

Equation (17) can be rewritten as an initial value problem as follows:

$$I' = [(t_f-t_0)/2]f(\tau), \qquad I(-1)=0 \tag{18}$$

where $\mathcal{I} = I(1)$. Equation (18) may be approximated via spectral differentiation. The initial condition is incorporated by eliminating the first row and column from the differentiation matrix $\boldsymbol{D}$. If $\tilde{\boldsymbol{D}}$ is the resulting $N\times N$ matrix, then we have

$$\boldsymbol{I} = [(t_f-t_0)/2]\tilde{\boldsymbol{D}}^{-1}\boldsymbol{f} \tag{19}$$

where $\boldsymbol{I} = \{I(\tau_1),\ldots,I(\tau_N)\}^T$, and $\boldsymbol{f} = \{f(\tau_1),\ldots,f(\tau_N)\}^T$. We are interested in the value of the integral at the end of the interval,

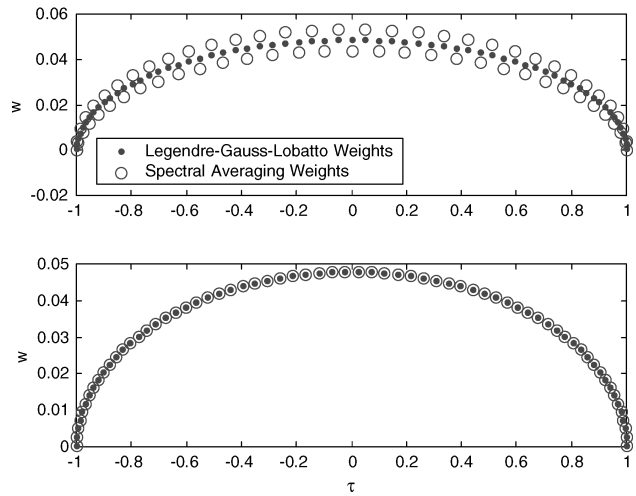

Fig. 1 Comparison of LGL weights and weights obtained from proposed discretization method: a) for N even (N = 64) and b) for N odd (N = 65).

which may be evaluated as

$$\mathcal{I} = I(\tau_N) = [(t_f - t_0)/2]\boldsymbol{v}^T\boldsymbol{f} \tag{20}$$

where $\boldsymbol{v} = \{v_1, \ldots, v_N\}^T$ is the last row of $\tilde{\boldsymbol{D}}^{-1}$. This method of integration ignores the value of $f(\tau)$ at $\tau = -1$, which can lead to unrealistic initial control inputs in an optimal control problem. To compensate for this effect, Eq. (17) can be identically written in the following form:

$$\mathcal{I} = \frac{1}{2}\frac{(t_f - t_0)}{2}\left(\int_{-1}^{1} f(\tau)\,\mathrm{d}\tau - \int_{1}^{-1} f(\tau)\,\mathrm{d}\tau\right) \tag{21}$$

The second integral term in Eq. (21) may be approximated in a similar fashion to the first term. This allows the original integral to be approximated by a finite sum,

$$\mathcal{I} \cong \frac{(t_f - t_0)}{2}\sum_{k=0}^{N} f(\tau_k)w_k \tag{22}$$

where

$$w_k = \begin{cases} -\tilde{v}_0/2 & k = 0 \\ (v_{k-1} - \tilde{v}_k)/2 & 1 \le k \le N-1 \\ v_{N-1}/2 & k = N \end{cases} \tag{23}$$

and $\tilde{\boldsymbol{v}} = \{\tilde{v}_0, \ldots, \tilde{v}_{N-1}\}^T$ is the first row of of the matrix $\tilde{\tilde{\boldsymbol{D}}}^{-1}$, where $\tilde{\boldsymbol{D}}$ is obtained by eliminating the last row and last column from the matrix $\boldsymbol{D}$. Figure 1 shows a comparison of the weights obtained by using Eq. (23) with the Legendre–Gauss–Lobatto (LGL) weights used in high-order quadrature formulas for $N = 64$ and $N = 65$. When N is odd, the LGL weights and the weights obtained from Eq. (23) agree very well, but when N is even, there is considerable difference between the two. Numerical experiments reveal that N should be odd for optimal approximation of integrals using the preceding method.

Discretization of the Optimal Control Problem

The optimal control problem reduces to the following discrete nonlinear programming problem: Find the coefficients

$$\boldsymbol{X} = [\boldsymbol{x}_0, \ldots, \boldsymbol{x}_N, \boldsymbol{u}_0, \ldots, \boldsymbol{u}_N]$$

to minimize

$$\mathcal{J}_N = \phi\left[\frac{4}{(t_f - t_0)^2}\boldsymbol{d}_N^{(2)}, \frac{2}{t_f - t_0}\boldsymbol{d}_N^{(1)}, \boldsymbol{x}_N, t_f\right] + \frac{(t_f - t_0)}{2}\sum_{k=0}^{N}\mathcal{L}\left(\frac{4}{(t_f - t_0)^2}\boldsymbol{d}_k^{(2)}, \frac{2}{t_f - t_0}\boldsymbol{d}_k^{(1)}, \boldsymbol{x}_k, \boldsymbol{u}_k, t_k\right)w_k \tag{24}$$

subject to

$$\boldsymbol{f}\left\{\left[4/(t_f - t_0)^2\right]\boldsymbol{d}_k^{(2)}, [2/(t_f - t_0)]\boldsymbol{d}_k^{(1)}, \boldsymbol{x}_k, \boldsymbol{u}_k, t_k\right\} = \boldsymbol{0} \qquad k = 0, \ldots, N \tag{25}$$

$$\boldsymbol{\psi}_0\left\{[2/(t_f - t_0)]\boldsymbol{d}_0^{(1)}, \boldsymbol{x}_0, t_0\right\} = \boldsymbol{0} \tag{26}$$

$$\boldsymbol{\psi}_f\left\{[2/(t_f - t_0)]\boldsymbol{d}_N^{(1)}, \boldsymbol{x}_N, t_f\right\} = \boldsymbol{0} \tag{27}$$

$$\boldsymbol{g}\left\{[2/(t_f - t_0)]\boldsymbol{d}_k^{(1)}, \boldsymbol{x}_k, \boldsymbol{u}_k\right\} \le \boldsymbol{0}, \qquad k = 0, \ldots, N \tag{28}$$

Equations (24–28) constitute a nonlinear programming problem that can be solved using standard optimization software such as NPSOL.[15]

Numerical Example

The control problem[6] is to find the thrust angle history $\gamma(t)$ to maximize the orbit radius $r(t_f)$ in a given time subject to

$$\dot{r} = u \tag{29}$$

$$\dot{\theta} = v/r \tag{30}$$

$$\dot{u} = v^2/r - 1/r^2 + [T/(1 - |\dot{m}|t)]\sin\gamma \tag{31}$$

$$\dot{v} = -uv/r + [T/(1 - |\dot{m}|t)]\cos\gamma \tag{32}$$

$$[r, \theta, u, v]_{t=0} = [1.0, 0, 0, 1.0] \tag{33}$$

$$[u, v]_{t=t_f} = \left[0, 1\Big/\sqrt{r(t_f)}\right] \tag{34}$$

where r is the orbit radius, θ is the orbit true anomaly, u is the radial component of the velocity, v is the tangential component of the velocity, t is the time, T is the thrust magnitude, γ is the thrust angle, and $-\dot{m}$ is the fuel consumption rate (constant). Note that canonical units are employed. Solutions are obtained for $T = 0.1405$, $|\dot{m}| = 0.0749$, and $t_f = 3.32$. As in Ref. 6, the initial guess is generated by numerically integrating the equations of motion with a constant value for the control of $\gamma(t) = 0.001$ rad. Solutions are obtained in MATLAB® 5.3 running in Windows XP on a Pentium 4 2.4-GHz using NPSOL[15] as the nonlinear solver. The Jacobians of the constraints are provided in analytical form.

This problem was solved using $N = 64$ for a variety of α and β. In each case, the performance index and computation time were recorded. The results are given in Table 1. The problem was also solved using a shooting method[16] and the case of $\alpha = \beta = 1$ is presented for comparison in Fig. 2.

Table 1 Comparison of cost function and computation times for orbit transfer problem

α	β	J	CPU time, s
−0.9999	−0.9999	−1.5252576	372.3
−0.99	−0.99	−1.5252608	373.4
−0.9	−0.9	−1.5252590	217.1
−0.8	−0.8	−1.5252569	432.9
−0.5	−0.5	−1.5252532	252.0
−0.3	−0.3	−1.5252530	613.1
0	0	−1.5252528	564.1
0.3	0.3	−1.5252529	290.6
0.5	0.5	−1.5252530	248.1
0.8	0.8	−1.5252533	274.1
1.0	1.0	−1.5252535	186.4
1.5	1.5	−1.5252547	146.2
2.0	2.0	−1.5252585	827.8
−0.9	0.9	−1.5252640	198.3
−0.7	0.7	−1.5252635	238.4
−0.5	0.5	−1.5252611	149.7
−0.3	0.3	−1.5252575	447.3
−0.1	0.1	−1.5252541	306.0
0.1	−0.1	−1.5252520	278.4
0.3	−0.3	−1.5252521	505.5
0.5	−0.5	−1.5252542	258.3
0.7	−0.7	−1.5252575	157.8
0.9	−0.9	−1.5252609	105.7

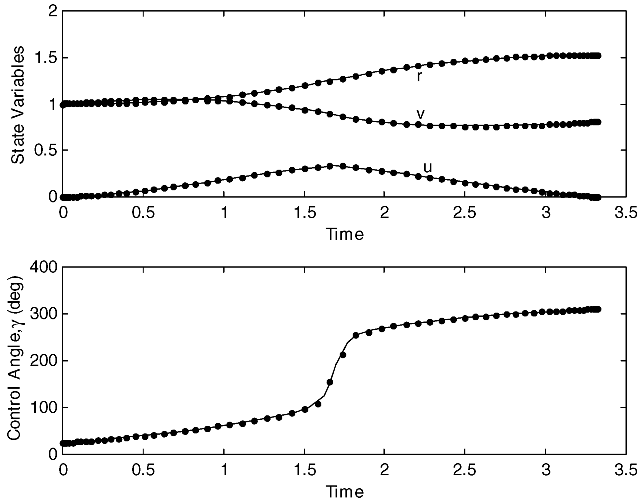

Fig. 2 a) Optimal states for orbit transfer problem and b) thrust direction: – – –, shooting solution and •, direct method with $\alpha=\beta=1$.

The results for the cost function in Table 1 are similar to those obtained by Fahroo and Ross.[6] Figure 2 shows the results for the case when $\alpha=\beta=1$ and clearly illustrate that the method converges to the optimal solution and that there is very close agreement with the solution from the shooting solution. Table 1 shows that there is only a small variation in the value of the cost as the value of α and β are varied (0.0008% maximum). However, it is observed that there is an extremely large variation in the computation time as α and β are varied (maximum 12.03 min). This has significant implications in choosing a suitable α and β for online computation of nonlinear optimal trajectories.

Conclusions

A general pseudospectral method based on Jacobi polynomials has been developed for solving optimal control problems. Numerical results indicate that significant differences in computation time occur as the parameters of the Jacobi polynomial vary. When the online calculation of optimal controls is considered, it may be important to choose the values of α and β that give the most efficient computation time for the specific application. Further work is necessary to address these issues.

Appendix A: Derivation of Lagrange Polynomials for JGL Nodes

The general expression for a Lagrange interpolating polynomial at N distinct nodes τ_k may be written as

$$\phi_j(\tau)=\prod_{\substack{k=0\\k\neq j}}^{N}\left(\frac{\tau-\tau_k}{\tau_j-\tau_k}\right)=\frac{g(\tau)}{(\tau-\tau_j)g'(\tau_j)},\qquad j=0,\ldots,N \tag{A1}$$

where

$$g(\tau)=\prod_{k=0}^{N}(\tau-\tau_k)=(\tau^2-1)P_N'^{(\alpha,\beta)}(\tau) \tag{A2}$$

$$g'(\tau_j)=\prod_{\substack{k=0\\k\neq j}}^{N}(\tau_j-\tau_k) \tag{A3}$$

by definition of the node points. The Jacobi polynomials $P_N^{(\alpha,\beta)}$ are eigenfunctions of the Sturm–Liouville equation

$$(1-\tau^2)P_N''^{(\alpha,\beta)}+[(\beta-\alpha)-(\alpha+\beta+2)\tau]P_N'^{(\alpha,\beta)}+N(N+\alpha+\beta+1)P_N^{(\alpha,\beta)}=0 \tag{A4}$$

It may be verified that

$$\left[(1-\tau^2)P_N'^{(\alpha,\beta)}\right]'=(1-\tau^2)P_N''^{(\alpha,\beta)}-2\tau P_N'^{(\alpha,\beta)} \tag{A5}$$

Differentiating Eq. (A2), and utilizing Eqs. (A4) and (A5), one obtains

$$g'(\tau)=[(\beta-\alpha)-(\alpha+\beta)\tau]P_N'^{(\alpha,\beta)}+N(N+\alpha+\beta+1)P_N^{(\alpha,\beta)} \tag{A6}$$

At the internal nodes τ_j, $1\le j\le N-1$, $P_N'^{(\alpha,\beta)}=0$ by definition of the nodes, and hence, from Eq. (A6),

$$g'(\tau_j)=N(N+\alpha+\beta+1)P_N^{(\alpha,\beta)}(\tau_j),\qquad j=1,\ldots,N-1 \tag{A7}$$

At the endpoints, $\tau_0=-1$ and $\tau_N=1$, Eq. (A4) can be used to obtain

$$P_N'^{(\alpha,\beta)}(\tau_0)=-\frac{N(N+\alpha+\beta+1)P_N^{(\alpha,\beta)}(\tau_0)}{2(\beta+1)} \tag{A8}$$

$$P_N'^{(\alpha,\beta)}(\tau_N)=\frac{N(N+\alpha+\beta+1)P_N^{(\alpha,\beta)}(\tau_N)}{2(\alpha+1)} \tag{A9}$$

Substitution of Eqs. (A8) and (A9) into Eq. (A6) yields

$$g'(\tau_0)=\frac{N(N+\alpha+\beta+1)P_N^{(\alpha,\beta)}(\tau_0)}{\beta+1} \tag{A10}$$

$$g'(\tau_N)=\frac{N(N+\alpha+\beta+1)P_N^{(\alpha,\beta)}(\tau_N)}{\alpha+1} \tag{A11}$$

Subtituting Eqs. (A7), (A10), and (A11) into Eq. (A1) gives the desired expression

$$\phi_j(\tau)=\frac{(\tau^2-1)}{(\tau-\tau_j)}\frac{c_jP_N'^{(\alpha,\beta)}(\tau)}{N(N+\alpha+\beta+1)P_N^{(\alpha,\beta)}(\tau_j)}\qquad j=0,\ldots,N \tag{A12}$$

where

$$c_j=\begin{cases}\beta+1, & j=0\\ 1, & 1\le j\le N-1\\ \alpha+1, & j=N\end{cases} \tag{A13}$$

Appendix B: Derivation of Jacobi Differentiation Matrix

In Appendix B, the derivation of the Jacobi differentiation matrix is outlined. The general expression for a Lagrange interpolating polynomial at N distinct nodes τ_k may be written as

$$\phi_j(\tau)=\frac{g(\tau)}{(\tau-\tau_j)g'(\tau_j)},\qquad j=0,\ldots,N \tag{B1}$$

Differentiating Eq. (B1) gives the entries of the differentiation matrix $\boldsymbol{D}$, whose entries are given by

$$\phi_j'(\tau_k)=\mathcal{D}_{kj}=\begin{cases}\dfrac{g'(\tau_k)}{(\tau-\tau_j)g'(\tau_j)}, & k\neq j\\[2ex] \dfrac{g''(\tau_j)}{2g'(\tau_j)}, & k=j\end{cases} \tag{B2}$$

where the diagonal elements are computed using L'Hospital's rule.

Off-Diagonal Elements

All of the off-diagonal elements are obtained by substituting Eqs. (A7), (A10), and (A11) into Eq. (B2).

Diagonal Elements

Differentiate Eq. (A6) to obtain

$$g''(\tau) = [(\beta - \alpha) - (\alpha + \beta)\tau] P_N^{\prime\prime(\alpha,\beta)} + [N(N + \alpha + \beta + 1) - (\alpha + \beta)] P_N^{\prime(\alpha,\beta)} \tag{B3}$$

At the internal nodes τ_j, $1 \le j \le N - 1$, $P_N^{\prime(\alpha,\beta)} = 0$ by definition of the nodes. Utilizing Eq. (A4), one obtains

$$P_N^{\prime\prime(\alpha,\beta)}(\tau_j) = \frac{N(N + \alpha + \beta + 1) P_N^{(\alpha,\beta)}(\tau_j)}{\left(\tau_j^2 - 1\right)} \tag{B4}$$

Substituting Eqs. (A7), (B3), and (B4) into Eq. (B2) yields

$$\phi_j'(\tau_j) = \frac{(\alpha + \beta)\tau_j + \alpha - \beta}{2\left(1 - \tau_j^2\right)}, \qquad j = 1, \ldots, N - 1 \tag{B5}$$

For the endpoint expressions, differentiate Eq. (A4) to obtain

$$P_N^{\prime\prime(\alpha,\beta)}(\tau_0) = \frac{[N(N + \alpha + \beta + 1) - (\alpha + \beta + 2)]}{2(\beta + 2)} \times \frac{N(N + \alpha + \beta + 1) P_N^{(\alpha,\beta)}(\tau_0)}{2(\beta + 1)} \tag{B6}$$

$$P_N^{\prime\prime(\alpha,\beta)}(\tau_N) = \frac{[N(N + \alpha + \beta + 1) - (\alpha + \beta + 2)]}{2(\alpha + 2)} \times \frac{N(N + \alpha + \beta + 1) P_N^{(\alpha,\beta)}(\tau_N)}{2(\alpha + 1)} \tag{B7}$$

Substituting Eqs. (B6) and (B7), together with Eqs. (A8) and (A9), into Eq. (B3) gives expressions for $g''(\tau_0)$ and $g''(\tau_N)$, which may be substituted into Eq. (B2) to give the remaining matrix elements.

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.

[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.

[3] 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.

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] Dontchev, A. L., Hager, W. W., and Veliov, V. M., "Second-Order Runge–Kutta Approximations in Control Constrained Optimal Control," *SIAM Journal of Numerical Analysis*, Vol. 38, No. 1, 2000, pp. 202–226.

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

[11] Fornberg, B., and Sloan, D. M., "A Review of Pseudospectral Methods for Solving Partial Differential Equations," *Acta Numerica*, Vol. 3, Aug. 1994, pp. 203–267.

[12] Gautschi, W., "High-Order Gauss–Lobatto Formulae," *Numerical Algorithms*, Vol. 25, No. 1, 2000, pp. 213–222.

[13] Ross, I. M., Rea, J., and Fahroo, F., "Exploiting Higher-Order Derivatives in Computational Optimal Control," *Proceedings of the MED 2002*, Mediterranean Conference on Control and Automation, Lisbon, Portugal, July 2002.

[14] Trefethen, L. N., *Spectral Methods in MATLAB*, SIAM Press, Philadelphia, 2000, pp. 124, 125.

[15] Gill, P. E., Murray, W., Saunders, M. A., and Wright, M. A., "User's Guide to NPSOL 5.0: A Fortran Package for Nonlinear Programming," Stanford Optimization Lab., Rept. TR SOL 86-1, Stanford Univ., Stanford, CA, July 1998.

[16] Bryson, A. E., *Dynamic Optimization*, Addison–Wesley, New York, 1999, Chap. 3.

Interferometric Observatories in Earth Orbit

I. I. Hussein,* D. J. Scheeres,† and D. C. Hyland‡
University of Michigan,
Ann Arbor, Michigan 48109-2140

I. Introduction

WE propose a class of satellite constellations that can act as interferometric observatories in Earth orbit. Based on techniques discussed in Refs. 1–3, the satellite constellation is capable of forming high-resolution images in timescales of a few hours without the need for active control beyond that needed for corrective maneuvers. First, we discuss the requirements to achieve these imaging goals. Next, we define a class of constellations that can achieve these goals. An optimization procedure is also defined that supplies m pixels of resolution with a minimum number of satellites. For the example considered, this procedure results in an observatory that is within 0–2 satellites from a lower bound of $\sqrt{m}$ satellites. The zonal J_2 effect is used to scan the observatory across the celestial sphere. Finally, we discuss the practical implementation of these observatories.

II. Imaging Requirements

Interferometric imaging is performed by measuring the mutual intensity (the two-point correlation[4]) that results from the collection and subsequent interference of two electric field measurements of a target made at two different observation points. While moving relative to each other, the satellites collect and transmit these measurements, which are later combined at a central node by use of precise knowledge of their locations and timing of data collection. A least-squares-error estimate of the image can be reconstructed given the mutual intensity measurements, parameters of the optical system, and the physical configuration of the observatory. To assess the quality of the reconstructed image, the reconstructed image is Fourier transformed into a two-dimensional plane of spatial frequencies (the wave number plane). At any given point on the wave number plane, the modulation transfer function (MTF) is defined as the ratio of the estimated intensity to the true image intensity. For an interferometric imaging constellation, the MTF can be

Received 4 April 2003; revision received 2 October 2003; accepted for publication 3 October 2003. Copyright © 2003 by the authors. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0731-5090/04 $10.00 in correspondence with the CCC.

*Ph.D. Candidate, Department of Aerospace Engineering; ihussein@umich.edu. Member AIAA.

†Associate Professor, Department of Aerospace Engineering; scheeres@umich.edu. Associate Fellow AIAA.

‡Professor and Chair, Department of Aerospace Engineering; dhiland@engin.umich.edu.