

A Gauss–Lobatto quadrature method for solving optimal control problems

P. Williams*

(Received 29 August 2005; revised 13 July 2006)

Abstract

This paper proposes a direct approach for solving optimal control problems. The time domain is divided into multiple subdomains, and a Lagrange interpolating polynomial using the Legendre–Gauss–Lobatto points is used to approximate the states and controls. The state equations are enforced at the Legendre–Gauss–Lobatto nodes in a nonlinear programming implementation by partial Gauss–Lobatto quadrature in each subdomain. The final state in each subdomain is enforced by a full Gauss–Lobatto quadrature. The Bolza cost functional is naturally approximated using Gauss–Lobatto quadrature across all subdomains.

*School of Aerospace, Mechanical, and Manufacturing Engineering, RMIT University, Bundoora, AUSTRALIA. <mailto:paul.williams@rmit.edu.au>

See <http://anziamj.austms.org.au/V47EMAC2005/Williams2> for this article, © Austral. Mathematical Soc. 2006. Published July 24, 2006. ISSN 1446-8735

Contents

1 Introduction	C102
2 Optimal control problem	C104
2.1 Integration matrix	C109
3 Numerical example	C113
4 Conclusions	C113
References	C114

1 Introduction

So-called direct methods have become extremely popular for solving optimal control problems, particularly in aerospace applications [1]. Trajectory optimization problems often involve both state and control constraints, as well as initial and terminal constraints. Two classes of solution methods are generally acknowledged for solving such problems: indirect and direct. Indirect methods are based on solving a two-point boundary value problem (TPBVP) that arises by applying Pontryagin’s Maximum Principle (PMP) to the original problem. The major difficulty in solving TPBVPs lies in the fact that the problem is ill-conditioned due to the extreme sensitivity of the costate dynamics. Generally, very good initial guesses and long computation times are required to obtain feasible solutions using these methods, particularly when there are state constraints. Direct methods, on the other hand, retain the structure of the original problem and discretize it directly. These methods have been termed direct collocation [2] or direct transcription [3]. The feature of these approaches is that both the states and controls are discretized at a selected number of collocation points or nodes. The state equations may be enforced by implicit integration rules such as the trape-

zoidal or Simpson rule [2], explicit integration rules such as Runge–Kutta [4], or by approximating the state history between the collocation points with Hermite interpolating polynomials and forcing the derivative of the polynomial to be equal to the state equations at internal collocation points [5]. Herman and Conway [5] showed that the latter method is equivalent to using Legendre–Gauss–Lobatto (LGL) quadrature rules for a particular choice of internal collocation points. The commonality in these methods is that the system dynamics are enforced only locally between the collocation points.

More recently, pseudospectral methods [6, 7] have been used for direct trajectory optimization. Contrary to the previous methods, pseudospectral methods are global in nature and approximate the entire state history with global Lagrange interpolating polynomials. The state equations are enforced at each of the nodes by differentiating the approximating polynomial and forcing it to be equal to the state equation evaluated at that point. Due to the global nature of the approximating polynomial, the differentiations can be achieved efficiently by employing a differentiation matrix. Spectral methods have been used widely for solving fluid dynamics problems and other boundary value problems [8].

In this paper, some of the above ideas are combined to form a pseudospectral method based purely on quadrature approximations of the underlying dynamical equations and cost function. A discretisation based on quadrature retains the advantage that linear differential equations are transformed into linear algebraic equations, which is not true of most typical direct methods. This allows linear quadratic optimal control problems to be solved without iteration. The quadrature-based approach is detailed in the following, and a numerical example is presented to illustrate some of the features of the method.

2 Optimal control problem

Consider the problem of finding the four-tuple $\{\mathbf{x}(t), \mathbf{u}(t), t_0, t_f\}$ to minimise the cost function

$$\mathcal{J} = \mathcal{E}[\mathbf{x}(t_f), t_f] + \int_{t_0}^{t_f} \mathcal{L}[\mathbf{x}(t), \mathbf{u}(t), t] dt, \quad (1)$$

subject to the nonlinear state equations

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \quad (2)$$

the end-point conditions

$$\mathbf{e}_L^0 \leq \mathbf{e}_0[\mathbf{x}(t_0), t_0] \leq \mathbf{e}_U^0, \quad (3)$$

$$\mathbf{e}_L^f \leq \mathbf{e}_f[\mathbf{x}(t_f), t_f] \leq \mathbf{e}_U^f, \quad (4)$$

the mixed state-control path constraints

$$\mathbf{g}_L \leq \mathbf{g}[\mathbf{x}(t), \mathbf{u}(t), t] \leq \mathbf{g}_U, \quad (5)$$

and box constraints

$$\mathbf{x}_L \leq \mathbf{x}(t) \leq \mathbf{x}_U, \quad \mathbf{u}_L \leq \mathbf{u}(t) \leq \mathbf{u}_U, \quad (6)$$

where $\mathbf{x} \in \mathbb{R}^{n_x}$ are the state variables, $\mathbf{u} \in \mathbb{R}^{n_u}$ are the control inputs, $t \in \mathbb{R}$ is the time, $\mathcal{E} : \mathbb{R}^{n_0} \times \mathbb{R} \rightarrow \mathbb{R}$ is the Mayer component of cost function, that is, the terminal, non-integral cost in Equation (1), $\mathcal{L} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \rightarrow \mathbb{R}$ is the Bolza component of the cost function, that is, the integral cost in Equation (1), $\mathbf{e}_L^0 \in \mathbb{R}^{n_x} \times \mathbb{R} \rightarrow \mathbb{R}^{n_0}$ and $\mathbf{e}_U^0 \in \mathbb{R}^{n_x} \times \mathbb{R} \rightarrow \mathbb{R}^{n_0}$ are the lower and upper bounds on the initial point conditions, $\mathbf{e}_L^f \in \mathbb{R}^{n_x} \times \mathbb{R} \rightarrow \mathbb{R}^{n_f}$ and $\mathbf{e}_U^f \in \mathbb{R}^{n_x} \times \mathbb{R} \rightarrow \mathbb{R}^{n_f}$ are the lower and upper bounds on the final point conditions, and $\mathbf{g}_L \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \rightarrow \mathbb{R}^{n_g}$ and $\mathbf{g}_U \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \rightarrow \mathbb{R}^{n_g}$ are the lower and upper bounds on the path constraints. The solution of general optimal control problems requires the application of the PMP to construct

the necessary conditions for optimality. The PMP essentially requires that the control Hamiltonian, defined as a function of “undetermined” covectors, be minimized.

Instead of employing the PMP to transform the problem into the higher dimensional state-costate space and using a discretisation method to transcribe the problem into a finite parameter set, it is far simpler to first transcribe the problem and apply the Karush–Kuhn–Tucker (KKT) conditions to the discrete problem. Nowadays, the KKT conditions are applied via nonlinear programming software without direct intervention of the user.

In the most general case (omitting switches in the states and/or vector field), it is convenient to divide the time domain into a series of M subdomains so that the time coordinates are

$$t^j \in I^j = [t_0 + (t_f - t_0)(j - 1)/M, t_0 + (t_f - t_0)j/M], \quad j = 1, \dots, M \quad (7)$$

in each subdomain. The fundamental idea behind using a Gauss–Lobatto quadrature discretisation is that the vector field is approximated by an N th degree Lagrange interpolating polynomial

$$\mathbf{f}(t) \approx \mathbf{f}^{N_j}(t), \quad t \in I^j. \quad (8)$$

For optimal interpolation approximation (in the l_2 -norm sense), the Lagrange interpolating polynomials are expanded using values of the vector field at a set of Legendre–Gauss–Lobatto (LGL) points. The LGL points are defined on the interval $[-1, 1]$ and correspond to the zeros of the derivative of the N th degree Legendre polynomial, $L_N(\tau)$, as well as the end points -1 and 1 . The Legendre polynomials are orthogonal to a unit weight function over the interval $\tau \in [-1, 1]$. The computational domain is related to the time domain by the affine transformation

$$t^j = [(t_f - t_0)\tau^j/M + 2t_0 + (t_f - t_0)(2j - 1)/M]/2, \quad j = 1, \dots, M. \quad (9)$$

In each subdomain, there are $N_j + 1$ LGL points. The Lagrange interpolating

polynomials take on the following form

$$\mathbf{f}^{N_j}(t^j) = \sum_{k=0}^{N_j} \mathbf{f}_k^j \phi_k^j(\tau^j), \quad (10)$$

where $t^j = t(\tau^j)$ because of the shifted computational domain. The Lagrange polynomials satisfy $\phi_k^j(\tau_i^j) = \delta_{ki}$, and hence the coefficients of the polynomial used in the expansion of the vector field take on their “physical” values at the LGL points. The Lagrange polynomials in Equation (10) may be expressed in terms of the Legendre polynomial basis functions

$$\phi_k^j(\tau) = \frac{(\tau^2 - 1) L'_{N_j}(\tau)}{(\tau - \tau_k^j) N_j (N_j + 1) L_{N_j}(\tau_k^j)}, \quad k = 0, \dots, N_j; \quad j = 1, \dots, M. \quad (11)$$

The vector field consists of values of the states and controls evaluated at the LGL points, and so the states and controls can themselves be considered as approximated via Lagrange polynomials similar to Equation (10). By integrating Equation (10) from the beginning of each subdomain, one obtains

$$\mathbf{x}_k^j = \mathbf{x}_0^j + \xi^j \int_{-1}^{\tau_k} \sum_{i=0}^{N_j} \phi_i^j(\tau^j) \mathbf{f}_i^j d\tau^j, \quad k = 1, \dots, N_j; \quad j = 1, \dots, M, \quad (12)$$

where $\xi^j = dt^j/d\tau^j = (t_f - t_0)/(2M)$. Define the entries of the $N \times (N + 1)$ integration matrix as $\mathcal{I}_{k-1,j} \triangleq \int_{-1}^{\tau_k} \phi_j(\tau^*) d\tau^*$, and noting that the coefficients \mathbf{f}_i^j are constant, allows Equation (12) to be expressed as

$$\mathbf{x}_k^j = \mathbf{x}_0^j + \xi^j \sum_{i=0}^{N_j} \mathcal{I}_{k-1,i} \mathbf{f}_i^j, \quad k = 1, \dots, N_j; \quad j = 1, \dots, M. \quad (13)$$

This discretisation may be recognised as a Gauss–Lobatto quadrature approximation to the state equations. The last row of the integration matrix corresponds to a full Gauss–Lobatto quadrature across the j th subdomain.

The discretisation of the cost function is performed using the full quadrature as

$$\mathcal{J}^N = \mathcal{E}(\mathbf{x}_{N_M}^M, t_f) + \sum_{j=1}^M \xi^j \sum_{i=0}^{N_j} \mathcal{L}(\mathbf{x}_i^j, \mathbf{u}_i^j, t_i^j) w_i^j, \quad (14)$$

where the Gauss–Lobatto quadrature weights are

$$w_k^j = \frac{2}{N_j(N_j + 1)} \frac{1}{[L_{N_j}(\tau_k^j)]^2}, \quad k = 0, \dots, N_j; \quad j = 1, \dots, M. \quad (15)$$

The end point constraints, path constraints, and box constraints are easily expressible in terms of the discrete decision variables

$$\mathbf{Z} \triangleq (\mathbf{x}_0^1, \dots, \mathbf{x}_{N_1}^1, \dots, \mathbf{x}_1^M, \dots, \mathbf{x}_{N_M}^M, \mathbf{u}_0^1, \dots, \mathbf{u}_{N_1}^1, \dots, \mathbf{u}_1^M, \dots, \mathbf{u}_{N_M}^M, t_0, t_f). \quad (16)$$

Note that the states and controls are implicitly constrained to be continuous across each subdomain so that $\mathbf{x}_{N_j}^j = \mathbf{x}_0^{j+1}$, and the extra variables are eliminated from the decision vector in Equation (16).

To illustrate an elegant property of the discretisation, consider the KKT conditions for the discretised problem. Note that a simplified version of the general problem is considered here for the sake of brevity (a global quadrature is assumed, and constraints are omitted).

The augmented cost is

$$\begin{aligned} \bar{\mathcal{J}}^N = \mathcal{E}(\mathbf{x}_N) + \tilde{\boldsymbol{\nu}}_0^\top \mathbf{e}_0(\mathbf{x}_0) + \tilde{\boldsymbol{\nu}}_f^\top \mathbf{e}_f(\mathbf{x}_N) + \xi \sum_{k=0}^N \mathcal{L}_k(\mathbf{x}_k, \mathbf{u}_k, t_k) w_k \\ + \sum_{k=1}^N \tilde{\boldsymbol{\lambda}}_k^\top \left(\xi \sum_{j=0}^N \mathcal{I}_{k-1,j} \mathbf{f}_j(\mathbf{x}_j, \mathbf{u}_j, t_j) + \mathbf{x}_0 - \mathbf{x}_k \right), \end{aligned} \quad (17)$$

where $\tilde{\boldsymbol{\nu}}_0$, $\tilde{\boldsymbol{\nu}}_f$, and $\tilde{\boldsymbol{\lambda}}_k$ are the KKT multipliers. The dual KKT conditions are obtained by taking partial derivatives of Equation (17) with respect to the

unknown parameters

$$\frac{\partial \bar{\mathcal{J}}^N}{\partial \mathbf{u}_i} = \xi \frac{\partial \mathcal{L}_i}{\partial \mathbf{u}_i} w_i + \xi \sum_{l=1}^N \mathcal{I}_{l-1,i} \left(\frac{\partial \mathbf{f}_i}{\partial \mathbf{u}_i} \right)^\top \tilde{\boldsymbol{\lambda}}_l = \mathbf{0}, \quad i = 0, \dots, N, \quad (18)$$

$$\begin{aligned} \frac{\partial \bar{\mathcal{J}}^N}{\partial \mathbf{x}_i} &= \xi \frac{\partial \mathcal{L}_i}{\partial \mathbf{x}_i} w_i + \xi \sum_{l=1}^N \mathcal{I}_{l-1,i} \left(\frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_i} \right)^\top \tilde{\boldsymbol{\lambda}}_l - \tilde{\boldsymbol{\lambda}}_i = \mathbf{0}, \\ & \quad i = 1, \dots, N-1, \end{aligned} \quad (19)$$

$$\begin{aligned} \frac{\partial \bar{\mathcal{J}}^N}{\partial \mathbf{x}_0} &= \left(\frac{\partial \mathbf{e}_0}{\partial \mathbf{x}_0} \right)^\top \tilde{\boldsymbol{\nu}}_0 + \xi \frac{\partial \mathcal{L}_0}{\partial \mathbf{x}_0} w_0 \\ & \quad + \xi \sum_{l=1}^N \mathcal{I}_{l-1,0} \left(\frac{\partial \mathbf{f}_0}{\partial \mathbf{x}_0} \right)^\top \tilde{\boldsymbol{\lambda}}_l + \sum_{l=1}^N \tilde{\boldsymbol{\lambda}}_l = \mathbf{0}, \end{aligned} \quad (20)$$

$$\begin{aligned} \frac{\partial \bar{\mathcal{J}}^N}{\partial \mathbf{x}_N} &= \frac{\partial \mathcal{E}}{\partial \mathbf{x}_N} + \left(\frac{\partial \mathbf{e}_f}{\partial \mathbf{x}_N} \right)^\top \tilde{\boldsymbol{\nu}}_f + \xi \frac{\partial \mathcal{L}_N}{\partial \mathbf{x}_N} w_N \\ & \quad + \xi \sum_{l=1}^N \mathcal{I}_{l-1,N} \left(\frac{\partial \mathbf{f}_N}{\partial \mathbf{x}_N} \right)^\top \tilde{\boldsymbol{\lambda}}_l - \tilde{\boldsymbol{\lambda}}_N = \mathbf{0}. \end{aligned} \quad (21)$$

Now define

$$\tilde{\boldsymbol{\Lambda}}_k = \sum_{l=1}^N \mathcal{I}_{l-1,k} \tilde{\boldsymbol{\lambda}}_l / w_k, \quad \bar{\boldsymbol{\Lambda}}_0 = - \left(\frac{\partial \mathbf{e}_0}{\partial \mathbf{x}_0} \right)^\top \tilde{\boldsymbol{\nu}}_0, \quad \bar{\boldsymbol{\Lambda}}_N = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_N} + \left(\frac{\partial \mathbf{e}_f}{\partial \mathbf{x}_N} \right)^\top \tilde{\boldsymbol{\nu}}_f.$$

Substitution into the control conditions gives

$$\frac{\partial \mathcal{L}_i}{\partial \mathbf{u}_i} + \tilde{\boldsymbol{\Lambda}}_i \left(\frac{\partial \mathbf{f}_i}{\partial \mathbf{u}_i} \right)^\top = 0, \quad i = 0, \dots, N, \quad (22)$$

which corresponds to a point-wise discretisation of the control conditions by application of the PMP. By rearranging and substituting subsequent equations with the remaining conditions,

$$\bar{\boldsymbol{\Lambda}}_0 - \xi \sum_{k=0}^N w_k \left[\frac{\partial \mathcal{L}_k}{\partial \mathbf{x}_k} + \tilde{\boldsymbol{\Lambda}}_k \left(\frac{\partial \mathbf{f}_k}{\partial \mathbf{x}_k} \right)^\top \right] = \bar{\boldsymbol{\Lambda}}_N, \quad (23)$$

which is recognised as a Gauss–Lobatto quadrature approximation to the continuous costate equation, including the boundary conditions. Note that a formal proof that the mapped covectors $\tilde{\Lambda}_k = \sum_{l=1}^N \mathcal{I}_{l-1,k} \tilde{\lambda}_l / w_k$ satisfy the same set of equivalent necessary conditions for optimality as the continuous time PMP conditions is beyond the scope of this work, but it can be seen that the mapped covectors derived above provide a good approximation to the continuous multipliers. This is supported by a range of numerical examples, including one presented in the numerical results section.

2.1 Integration matrix

The general expression for the Lagrange interpolating polynomial is

$$\phi_k(\tau) = \frac{g(\tau)}{(\tau - \tau_k) g'(\tau_k)}, \quad (24)$$

and the entries of the integration matrix are

$$\mathcal{I}_{i-1,k} = \int_{-1}^{\tau_i} \phi_k(\tau) d\tau. \quad (25)$$

By definition of the LGL nodes,

$$g(\tau) = (\tau^2 - 1) L'_N(\tau). \quad (26)$$

It may also be shown that

$$(\tau^2 - 1) L'_N(\tau) = \frac{N(N+1)}{(2N+1)} [L_{N+1}(\tau) - L_{N-1}(\tau)]. \quad (27)$$

Hence, by the properties of the Legendre polynomial,

$$g'(\tau) = \frac{N(N+1)}{(2N+1)} [L'_{N+1}(\tau) - L'_{N-1}(\tau)] = N(N+1) L_N(\tau). \quad (28)$$

Using the Christoffel–Darboux identity [9]

$$B_m = \sum_{k=0}^m \frac{L_k(\tau) L_k(z)}{\gamma_k} = \frac{L_{m+1}(\tau) L_m(z) - L_m(\tau) L_{m+1}(z)}{A_{m+1} \gamma_m (\tau - z) / A_m}, \quad (29)$$

where for the Legendre polynomials,

$$\gamma_m = \frac{2}{2m + 1}, \quad A_m = \frac{(2m)!}{2^m (m!)^2}, \quad \text{and hence} \quad \frac{A_{m+1} \gamma_m}{A_m} = \frac{2}{m + 1};$$

also

$$B_N(\tau, z) + \frac{N + 1}{N} B_{N-1}(\tau, z) = \frac{(2N + 1)}{2N} \left[\frac{g(\tau) L_N(z) - L_N(\tau) g(z)}{\tau - z} \right]. \quad (30)$$

Substituting $z = \tau_k$, where τ_k is a zero of $g(\tau)$, we are led to

$$B_N(\tau, \tau_k) + \frac{N + 1}{N} B_{N-1}(\tau, \tau_k) = \frac{(2N + 1)}{2N} \left[\frac{g(\tau) L_N(\tau_k)}{\tau - \tau_k} \right]. \quad (31)$$

Integrating the RHS of Equation (31) leads to

$$\begin{aligned} & \int_{-1}^{\tau_i} \frac{(2N + 1)}{2N} \left[\frac{g(\tau) L_N(\tau_k)}{\tau - \tau_k} \right] d\tau \\ &= \frac{(2N + 1)}{2N} \mathcal{I}_{i-1,k} g'(\tau_k) L_N(\tau_k) \\ &= \frac{(2N + 1)}{2} (N + 1) [L_N(\tau_k)]^2 \mathcal{I}_{i-1,k}. \end{aligned} \quad (32)$$

Making use of the identity

$$L_{N+1}(\tau) - L_{N-1}(\tau) = (2N + 1) \int_{-1}^{\tau} L_N(\tau^*) d\tau^* \quad (33)$$

to integrate the LHS of Equation (31) leads to, for example for the first term,

$$\int_{-1}^{\tau_i} B_N(\tau, \tau_k) d\tau = \int_{-1}^{\tau_i} \sum_{j=0}^N \frac{L_j(\tau_k) L_j(\tau)}{\gamma_j} d\tau$$

$$= \frac{1}{2} \sum_{j=0}^N L_j(\tau_k) [L_{j+1}(\tau_i) - L_{j-1}(\tau_i)]. \quad (34)$$

Finally, the LHS

$$\begin{aligned} & \int_{-1}^{\tau_i} \left[B_N(\tau, \tau_k) + \frac{N+1}{N} B_{N-1}(\tau, \tau_k) \right] d\tau \\ &= \left(\frac{2N+1}{2N} \right) \left\{ 1 + \tau_i + \sum_{j=1}^N L_j(\tau_k) [L_{j+1}(\tau_i) - L_{j-1}(\tau_i)] \right\}, \end{aligned} \quad (35)$$

which makes use of

$$[L_{N+1}(\tau_i) - L_{N-1}(\tau_i)] = 0, \quad (36)$$

and

$$L_0(\tau_k) [L_1(\tau_i) - L_{-1}(\tau_i)] = \tau_i + 1. \quad (37)$$

Rearranging the expressions in Eqs. (32) and (35) gives

$$\begin{aligned} \mathcal{I}_{i-1,k} &= \frac{w_k}{2} \left\{ 1 + \tau_i + \sum_{j=1}^N L_j(\tau_k) [L_{j+1}(\tau_i) - L_{j-1}(\tau_i)] \right\}, \\ & \quad i = 1, \dots, N; \quad k = 0, \dots, N. \end{aligned} \quad (38)$$

The Legendre–Gauss–Lobatto quadrature weights are recovered for $\tau_i = 1$. This formula has been validated by using a modified Chebyshev moments method to solve for each row of the integration matrix numerically.

The LGL nodes, which are theoretically the roots of the derivative of the N th degree Legendre polynomial, including the end points, should not be computed by a root-finding algorithm. It is much more accurate and efficient to compute them as the eigenvalues of a modified tridiagonal Jacobi matrix as discussed by Gautschi [10] for the general Jacobi case.

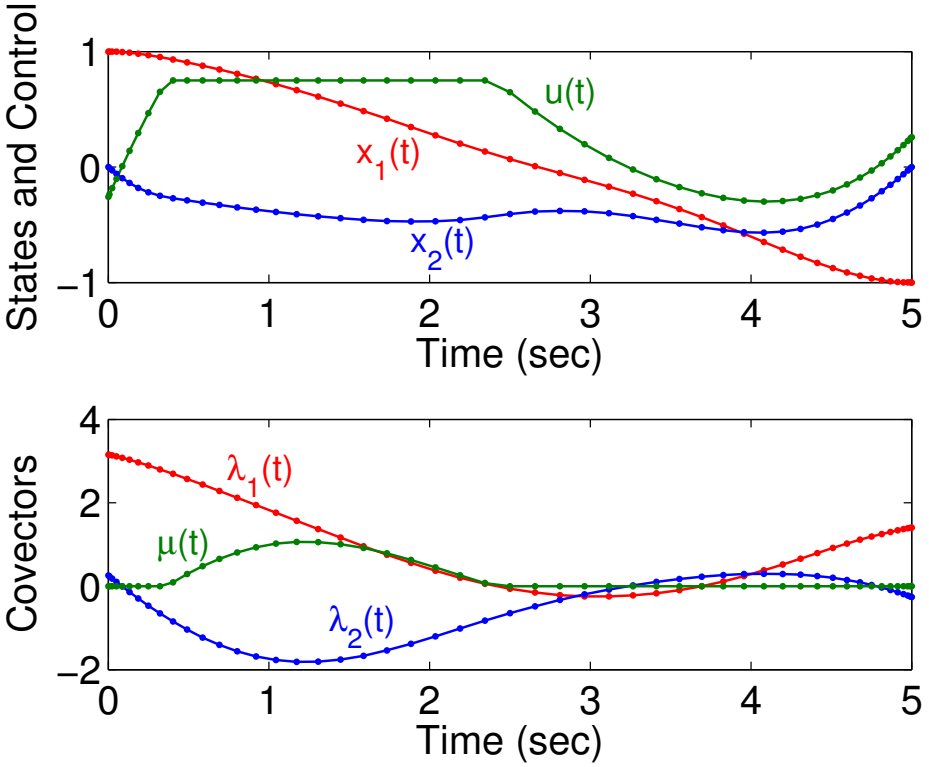


FIGURE 1: Numerical results for Van der Pol problem, (—) indirect, (···) direct

3 Numerical example

To illustrate the features of the method, consider a constrained Van der Pol oscillator problem. The control problem is to minimise the cost

$$\mathcal{J} = \frac{1}{2} \int_0^5 (x_1^2 + x_2^2 + u^2) dt, \quad (39)$$

subject to the constraints

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = -x_1 + (1 - x_1^2)x_2 + u, \quad u(t) \leq 0.75, \quad (40)$$

and the boundary conditions $x_1(0) = 1$, $x_2(0) = 0$, $x_1(5) = -1$, $x_2(5) = 0$. This is a fixed-time, constrained optimal control problem with initial and terminal constraints (two-point boundary value problem). This problem is solved using an indirect quasilinearisation method as well as the proposed method with one interval and $N = 50$. The D -form of the Lagrangian of the Hamiltonian for this problem is

$$L = \frac{1}{2} (x_1^2 + x_2^2 + u^2) + \lambda_1 x_2 + \lambda_2 [(1 - x_1^2)x_2 - x_1 + u] + \mu (u - 0.75),$$

which allows the necessary conditions to be determined in a straightforward manner. The direct problem is solved using the sequential quadratic programming software SNOPT. Numerical results are shown in Figure 1. There is excellent agreement in both the primal and dual solutions for this problem. Figure 1 illustrates that the terminal constraints are met precisely.

4 Conclusions

A direct approach for the solution of nonlinear optimal control problems has been developed utilising Gauss–Lobatto quadrature to formulate the nonlinear constraint equations and cost function. The approach is implemented

efficiently using an integration matrix, whose entries are analytically derived. A set of mapped covectors approximate the continuous covectors for the optimal control problem, and a numerical example illustrates excellent agreement with an indirect method.

References

- [1] Betts, J. T., Survey of numerical methods for trajectory optimization, *Journal of Guidance, Control, and Dynamics*, **21**, 1998, 193–207.
<http://pdf.aiaa.org/jaPreview/JGCD/1998/PVJAIMP4231.pdf>
C102
- [2] Hargraves, C. R., and Paris, S. W., Direct trajectory optimization using nonlinear programming and collocation, *Journal of Guidance, Dynamics, and Control*, **10**, 1987, 338–342.
<http://pdf.aiaa.org/jaPreview/JGCD/1987/PVJAPRE20223.pdf>
C102, C103
- [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*, **15**, 1992, 994–1002.
<http://pdf.aiaa.org/jaPreview/JGCD/1992/PVJAPRE20934.pdf>
C102
- [4] Hager, W. W., Runge–Kutta method in optimal control and the transformed adjoint system, *Numerische Mathematik*, **87**, 2000, 247–282. <http://dx.doi.org/10.1007/s002110000178> C103
- [5] 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*, **19**, 1996, 592–599.
<http://pdf.aiaa.org/jaPreview/JGCD/1996/PVJAPRE21662.pdf>
C103

- [6] Elnagar, G., Kazemi, M. A., and Razzaghi, M., The pseudospectral legendre method for discretizing optimal control problems, *IEEE Transactions on Automatic Control*, **40**, 1995, 1793–1796.
<http://dx.doi.org/10.1109/9.467672> C103
- [7] Ross, I. M., and Fahroo, F., Legendre pseudospectral approximations of optimal control problems, *Lecture Notes in Control and Information Sciences*, **295**, Springer-Verlag, Berlin, 2003.
<http://dx.doi.org/10.1007/b80168> C103
- [8] Fornberg, B., and Sloan, D. M., A review of pseudospectral methods for solving partial differential equations, *Acta Numerica*, **3**, 1994, 203–267. C103
- [9] Hildebrand, F. B., *Introduction to Numerical Analysis, Second Edition*, McGraw-Hill, New York, 1974. C110
- [10] Gautschi, W., High-order Gauss–Lobatto formulae, *Numerical Algorithms*, **25**, 2000, 213–222.
<http://dx.doi.org/10.1023/A:1016689830453> C111