A Multidimensional Pseudospectral Method for Optimal Control

Justin Ruths

May 20, 2010

1 Optimal Control

Consider the general optimal control problem

$$\begin{aligned} \min \quad & \varphi(T, x(T)) + \int_0^T \mathcal{L}(x(t), u(t))\, dt \\ \text{s.t.} \quad & \frac{d}{dt}x(t) = f(x(t), u(t)), \\ & e(x(0), x(T)) = 0, \\ & g(x(t), u(t)) \leq 0, \end{aligned} \tag{1}$$

where φ and $\mathcal{L}$ are the terminal (evaluated at the terminal time, T) and running cost (depending on the time history of the state and control variables) terms of the general objective function, respectively; f is the system dynamics; e represents endpoint constraints, and g denotes path constraints.

Solving this optimal control problem in the continuous time domain analytically is generally intractable for most complicated systems of interest and we, therefore, turn to computational methods to derive numerical solutions. Our goal is to transform this optimal control problem defined on a function space to a constrained optimization on a vector space.

2 Pseudospectral Method

The overarching goal of the pseudospectral method is to convert the continuous optimal control problem in (1) into a constrained algebraic minimization problem, which can be solved by efficient nonlinear numerical optimization solvers. The pseudospectral method was originally developed to solve problems in fluid dynamics and since then has been successfully applied to many areas of science and engineering [2, 3, 4].

Pseudospectral discretization methods use expansions of orthogonal polynomials (see Appendix A) to approximate the states of the system and thereby inherit the spectral accuracy characteristic of orthogonal polynomial expansions (the k^{th} coefficient of the expansion decreases faster than any inverse power of k) [5]. Through special recursive properties (see Appendix B), derivatives of these orthogonal polynomials can be expressed in terms of the polynomials themselves, making it possible to accurately approximate the differential equation that describes the dynamics with an algebraic relation imposed at a small number of discretization points. An appropriate choice of these discretization points, or nodes, facilitates the approximation of the states as well as ensuring accurate numerical integration through Gaussian quadrature.

We first transform the original problem from the time domain $t \in [0, T]$ to the rescaled domain $t \in [-1, 1]$ on which the orthogonal polynomials are defined. Our choice of the Legendre orthogonal polynomial family suggests we compute the integral term of the cost function using Legendre-Gauss-Lobatto (LGL) quadrature, in which the integral is approximated by a summation of the integrand evaluated at specific set of nodes,

$$\int_{-1}^{1} f(t)dt \approx \sum_{i=1}^{N} f(t_i)w_i, \qquad w_i = \int_{-1}^{1} \ell_i(t)dt, \tag{2}$$

where N is the order of polynomial approximation, w_i are discrete weights, and $\ell_i(t)$ is the i^{th} Lagrange polynomial, discussed below and in Appendix C [7]. Lobatto in LGL refers to the inclusion of the endpoints as nodes, which is necessary in discretizing optimal control problems in order to enforce initial and terminal conditions. In particular, if the integrand $f \in \mathbb{P}_{2N-1}$ and the nodes $t_i \in \Gamma^{\text{LGL}}$, the integral approximation is exact, where $\mathbb{P}_{2N-1}$ denotes the set of polynomials of degree $2N-1$ or less and where $\Gamma^{\text{LGL}} = \{t_i : L'_N(t)|_{t_i} = 0, i = 1, \dots N-1\} \bigcup \{-1, 1\}$ are the $N+1$ LGL nodes determined by the derivative of the N^{th} order Legendre polynomial, $L'_N(t)$, and the interval endpoints [5].

LGL quadrature requires we know the integrand values at the LGL nodes, however, the N^{th} order Legendre expansions

$$x(t) \approx P_N x(t) = \textstyle\sum_{k=0}^{N} \tilde{x}_k L_k(t), \tag{3}$$

$$u(t) \approx P_N u(t) = \textstyle\sum_{k=0}^{N} \tilde{u}_k L_k(t), \tag{4}$$

do not directly give us a way to discretize the states and controls at these nodes, i.e. the expansions coefficients $\tilde{x}_k$ and $\tilde{u}_k$ have no direct physical meaning. To overcome this, we approximate these Legendre expansions with interpolating polynomials, which, by definition, are equal to the Legendre expansions at the interpolation nodes. Because any interpolating polynomial can be represented by Lagrange polynomials we can represent the state and control as,

$$P_N x(t) \approx I_N x(t) = \textstyle\sum_{k=0}^{N} \bar{x}_k \ell_k(t), \tag{5}$$

$$P_N u(t) \approx I_N u(t) = \textstyle\sum_{k=0}^{N} \bar{u}_k \ell_k(t), \tag{6}$$

where the coefficients $\bar{x}_k$ and $\bar{u}_k$ are the values of the state and control Legendre expansions evaluated at the k^{th} interpolation node, respectively, i.e., $P_N x(t_k) = I_N x(t_k) = \bar{x}_k$ and $P_N u(t_k) = I_N u(t_k) = \bar{u}_k$. The coefficients have this property because the k^{th} Lagrange polynomial is characterized by taking unit value at the k^{th} interpolation node and zero value at all other nodes such that $\ell_k(t_i) = \delta_{ki}$, where δ_{ki} is the Kronecker delta function [6]. Using this second approximation we can compute the integrand of the cost function integral at the LGL nodes and $\bar{x}_k$ and $\bar{u}_k$ become the decision variables of the subsequent discrete problem.

Furthermore, the selection of LGL nodes, which are non-uniform on $[-1, 1]$ with quadratic spacing towards the endpoints, as interpolation nodes suppresses the spurious oscillations between nodes that is present when using uniformly spaced nodes, called the Runge phenomena [8]. It can be shown that the LGL interpolation nodes are close to optimal (see Appendix D). The LGL nodes permit us rewrite the Lagrange polynomials in terms of the Legendre polynomials, which is critical to inherit the special derivative and spectral accuracy properties of the orthogonal polynomials despite using Lagrange interpolating polynomials. Given $t_k \in \Gamma^{\text{LGL}}$, we can express the Lagrange polynomial as (see Appendix E) [10],

$$\ell_k(t) = \frac{1}{N(N+1)L_N(t_k)} \frac{(t^2-1)L'_N(t)}{t - t_k}. \tag{7}$$

The derivative of (5) at $t_i \in \Gamma^{\text{LGL}}$ is then,

$$\begin{aligned} \frac{d}{dt} I_N x(t_i) &= \sum_{k=0}^{N} \bar{x}_k \dot{\ell}_k(t_i) = \sum_{k=0}^{N} D_{ik} \bar{x}_k \\ &= D_{i0}\bar{x}_0 + D_{i1}\bar{x}_1 + \cdots + D_{iN}\bar{x}_N, \end{aligned} \tag{8}$$

where D is the constant matrix with elements (see Appendix F) [11],

$$D_{ik} = \begin{cases} \frac{L_N(t_i)}{L_N(t_k)} \frac{1}{t_i - t_k} & i \neq k \\ -\frac{N(N+1)}{4} & i = k = 0 \\ \frac{N(N+1)}{4} & i = k = N \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

We have now effectively discretized all parts of the original optimal control problem. The problem in (1) can now be written as,

$$\begin{aligned} \min \quad & \varphi(T, \bar{x}_N) + \frac{T}{2}\sum_{i=0}^{N} \mathcal{L}(\bar{x}_i, \bar{u}_i) w_i \\ \text{s.t.} \quad & \sum_{k=0}^{N} D_{ik}\bar{x}_k = \frac{T}{2} f(\bar{x}_i, \bar{u}_i), \\ & e(\bar{x}_0, \bar{x}_N) = 0, \\ & g(\bar{x}_i, \bar{u}_i) \leq 0, \quad \forall\, i \in \{0, 1, \ldots, N\}. \end{aligned} \tag{10}$$

Notice that the second and third lines are equality constraints reflecting the dynamics and endpoint conditions, respectively, and the last line is an inequality constraint reflecting the path constraints.

3 Optimal Ensemble Sampling

The ensemble optimal control problem in (11) includes another dimension of continuity in the parameter domain, $s \in \Omega \subset \mathbb{R}^{\mathrm{d}}$, which must be discretized (or sampled) to fit within the constrained minimization method. To reduce the complexity of notation we consider only a single parameter variation, i.e., $d = 1$ and $\Omega = [\underline{s}, \bar{s}]$, however, it is straightforward to extend this to higher dimensions.

$$\begin{aligned} \min \quad & \int_{\Omega} \Big[\varphi(T, x(T,s)) + \int_0^T \mathcal{L}(x(t,s), u(t))\, dt\Big] ds \\ \text{s.t.} \quad & \frac{d}{dt} x(t,s) = f(x(t,s), u(t)), \\ & e(x(0,s), x(T,s)) = 0, \\ & g(x(t,s), u(t)) \leq 0, \end{aligned} \tag{11}$$

Consider now the ensemble extension of the interpolation approximation in (5),

$$\begin{aligned} x(t,s) \approx I_{N\times N_s} x(t,s) &= \sum_{k=0}^{N} \bar{x}_k(s)\ell_k(t) \\ &\approx \sum_{k=0}^{N} \left(\sum_{r=0}^{N_s} \bar{x}_{kr}\ell_r(s)\right) \ell_k(t), \end{aligned} \tag{12}$$

and the ensemble extension of the approximate derivative from (8) at $t_i \in \Gamma^{\mathrm{LGL}}$ and $s_j \in \Gamma^{\mathrm{LGL}}_{N_s}$,

$$\begin{aligned} \frac{d}{dt} I_{N\times N_s} x(t_i, s_j) &= \sum_{k=0}^{N} D_{ik} \left(\sum_{r=0}^{N_s} \bar{x}_{kr}\ell_r(s_j)\right) \\ &= \sum_{k=0}^{N} D_{ik}\bar{x}_{kj}, \end{aligned} \tag{13}$$

where $\bar{x}_{kj} = x(t_k, s_j)$. In (12) and (13) we have effectively used a two dimensional interpolating grid at the $N+1$ and N_s+1 LGL nodes in time and the parameter, respectively. Using (12), (13), in conjunction with the LGL quadrature rule, we summarize the ensemble pseudospectral discretization of the ensemble optimal control problem as

$$
\begin{aligned}
\min \quad & \frac{\bar{s}-\underline{s}}{2} \sum_{r=0}^{N_s} \Big[\varphi(T, \bar{x}_{Nr}) + \frac{T}{2} \sum_{i=0}^{N} \mathcal{L}(\bar{x}_{ir}, \bar{u}_{ir}) w_i^N \Big] w_r^{N_s} \\
\text{s.t.} \quad & \sum_{k=0}^{N} D_{ik} \bar{x}_{kr} = \frac{T}{2} f(\bar{x}_{ir}, \bar{u}_i), \\
& e(\bar{x}_{0r}, \bar{x}_{Nr}) = 0, \\
& g(\bar{x}_{ir}, \bar{u}_{ir}) \leq 0, \quad \forall \begin{array}{c} i \in \{0,1,\ldots,N\} \\ r \in \{0,1,\ldots,N_s\} \end{array},
\end{aligned}
\tag{14}
$$

where w^N and w^{N_s} are the LGL quadrature weights corresponding to polynomial approximations of order N and N_s respectively. Notice that the summation across the parameter domain in the cost function is multiplied by the interval length. This is because the summation reflects the integral, which is defined on $[-1, 1]$.

4 Implementation

Now that the pseudospectral method transforms a continuous optimal control problem to a constrained minimization on a vector space, the problem becomes much more straightforward to solve numerically. We focus here on the AMPL modeling language, however, this can be done in virtually any environment with linear algebra routines and nonlinear optimization capability. AMPL provides a succinct and powerful way of coding these problems which enables them to be input quickly and new problems to be easily adapted from older problems. At the same time AMPL is a gateway to many solvers making it also a powerful optimization choice.

The original problem has unknown functions $x(t) \in \mathbb{R}^n$ (or $x(t, s)$ in the ensemble case) and $u(t) \in \mathbb{R}^m$ and a possibly unknown terminal time T. Approximating the states and controls with polynomials of order N on the nodes Γ_N^{LGL} creates interpolating polynomials with coefficients $\bar{x}_i$ and $\bar{u}_i$

$$
\left[\begin{array}{c} x_1(t) \\ \vdots \\ x_n(t) \\ \hline u_1(t) \\ \vdots \\ u_m(t) \end{array} \right]
\Rightarrow
\left[\begin{array}{cccc} \bar{x}_{10} & \bar{x}_{11} & \cdots & \bar{x}_{1N} \\ \vdots & & & \vdots \\ \bar{x}_{n0} & \bar{x}_{n1} & \cdots & \bar{x}_{nN} \\ \hline \bar{u}_{10} & \bar{u}_{11} & \cdots & \bar{u}_{1N} \\ \vdots & & & \vdots \\ \bar{u}_{m0} & \bar{u}_{m1} & \cdots & \bar{u}_{mN} \end{array} \right]
$$

which become the decision variables of the discretized optimization in addition to the terminal time T.

4.1 AMPL Syntax

There are only a few key concepts which compose AMPL code. AMPL code is broken into two sections, the model and the data. The model typically describes the form of the problem, but no specific values and leaves these to be specified in the data section. The model is composed of parameters (`param`), variables (`var`), collections on which to define parameters and variables (`set`), a quantity to maximize or minimize (`maximize` or `minimize`), and constraints (`subject to`). These are followed by `data;` and then specifications of values for the parameters and possibly variables for the optimization. There are many excellent AMPL references freely available online. See Appendix G for examples.

A Orthogonal Polynomials

Given a non-negative weight function $w(t) \geq 0$, $\int_a^b w(t)dt > 0$, and a weighted inner product $f, g \in L^2_w(a, b)$,

$$\langle f, g \rangle_w = \int_a^b f(t)g(t)w(t)dt,$$

it is possible to create an orthogonal basis, $\{\phi_k\}$, using the Gram-Schmidt process, i.e.,

$$\langle \phi_i, \phi_j \rangle_w \propto \delta_{ij}.$$

Furthermore, orthogonalizing the non-negative powers of t yields a set of orthogonal *polynomials*,

$$(1, t, t^2, \ldots, t^N; w(t)) \Rightarrow \{p_k\}$$

where $p_k \in \mathbb{P}_{\mathbb{N}}$. Legendre polynomials, $\{L_k(t)\}$, are derived with unit weight function, $w(t) = 1$. Therefore, the Gram-Schmidt process for the Legendre polynomials is given by $L_0(t) = 1$ and

$$L_k(t) = t^k - \sum_{i=0}^{k-1} \underbrace{\frac{\langle t^k, L_i(t) \rangle}{\langle L_i(t), L_i(t) \rangle}}_{\text{project } t^k \text{ onto } L_i(t)} L_i(t) = t^k - \sum_{i=0}^{k-1} \frac{\int_{-1}^1 t^k L_i(t)dt}{\int_{-1}^1 L_i^2(t)dt} L_i(t)$$

for $k > 0$. A different orthogonal polynomial family would use a different weight, w, but the process would be similar.

B Legendre Polynomial Properties for Optimal Control

Recall that the ability of spectral methods to convert a differential equation into an algebraic equation is the feature which makes them powerful tools for problems such as those of optimal control. Legendre polynomials, $L_k(t)$, obey a recursion relation,

$$L_{k+1}(t) = \frac{2k+1}{k+1} t L_k(t) - \frac{k}{k+1} L_{k-1}(t) \tag{15}$$

and also the differential relation,

$$[(1 - t^2)L'_k(t)]' + k(k+1)L_k(t) = 0 \tag{16}$$

These two relations illustrate how for function x expanded in terms of L_k, we can express $x'(t) = \dot{x}$ in terms of L_k as well - rather than in terms of L'_k. Hence, $\dot{x} = f(x)$ is now an algebraic equation since both sides can be written as an expansion using L_k as basis functions. Other useful properties of Legendre polynomials are

$$L_k(\pm 1) = (\pm 1)^k \tag{17}$$

$$L'_k(\pm 1) = \frac{(\pm 1)^{k+1} k(k+1)}{2} \tag{18}$$

C Lagrange Interpolating Polynomials

Any interpolating polynomial can be represented by the Lagrange polynomial basis. The k^{th} Lagrange polynomial is characterized by taking unit value at the k^{th} interpolation node and zero at all other nodes, which is effectively a shifted Kronecker delta function, i.e. $\ell_k(t_i) = \delta_{ki}$. The Lagrange polynomials can be written in several ways, but the most transparent is the following fractional product of the interpolation nodes,

$$\ell_k(t) = \frac{(t-t_0)\cdots(t-t_{k-1})(t-t_{k+1})\cdots(t-t_N)}{(t_k-t_0)\cdots(t_k-t_{k-1})(t_k-t_{k+1})\cdots(t_k-t_N)} \tag{19}$$

$$= \prod_{\substack{i=0\\ i\neq k}}^{N} \frac{(t-t_i)}{(t_k-t_i)} \ . \tag{20}$$

D Optimal Interpolation Nodes

The optimality of a specific choice of interpolation nodes can been quantified by

$$||x - I_N x||_\infty \leq (1+\Lambda_N(\Gamma))||x - p_N^*(x)||_\infty \ , \tag{21}$$

where $p_N^*(x)$ is the best approximating polynomial with respect to the uniform norm and $\Lambda_N(\Gamma)$ is the Lebesgue constant defined by

$$\Lambda_N(\Gamma) = \max_{t\in[-1,1]} \sum_{k=0}^{N} |\ell_k(t)| \ , \tag{22}$$

with $\ell_k(t)$ the k^{th} Lagrange polynomial for the interpolation grid Γ. The Lebesgue constant, then, gives the maximum cumulative excursion from zero of the Lagrange polynomial family along the time axis. Although a closed form for the Lebesgue constant is not in the literature, as $N \to \infty$ the Chebychev-Gauss grid (close to the LGL nodes) yields,

$$\Lambda_N(\Gamma_{\text{CG}}) = \frac{2}{\pi}\log N + \frac{2}{\pi}\Big(\gamma + \log\frac{8}{\pi} - \frac{2}{3}\Big) + O\Big(\frac{1}{\log N}\Big)$$

which in this limit is asymptotic to the Lebesgue constant of the optimal interpolation grid [9],

$$\Lambda_N(\Gamma_{\text{CG}}) = \frac{2}{\pi}\log N + \frac{2}{\pi}\Big(\gamma + \log\frac{4}{\pi}\Big) + O\Big(\Big(\frac{\log\log N}{\log N}\Big)^2\Big).$$

E Lagrange Polynomial written in terms of Legendre Polynomial

Define $w(t) = \prod_{i=0}^{N}(t-t_i)$. Taking the derivative,

$$w'(t) = \sum_{k=0}^{N}\prod_{\substack{i=0\\ i\neq k}}^{N}(t-t_i) \qquad \Rightarrow \qquad w'(t_k) = \prod_{\substack{i=0\\ i\neq k}}^{N}(t_k-t_i) \tag{23}$$

We can now express (19) from Appendix C as,

$$\ell_k(t) = \frac{w(t)}{(t-t_k)w'(t_k)} \tag{24}$$

Recall that the LG nodes (LGL nodes excluding the endpoints) $\{t_1,\ldots,t_{N-1}\}$ are zeros of $L_N'(t)$, therefore $L_N'(t) = (t-t_1)\ldots(t-t_{N-1})$. We can then write $w(t)$ in terms of the N degree Legendre polynomial.

$$w(t) = (t-t_0)\underbrace{(t-t_1)\ldots(t-t_{N-1})}_{L_N'(t)}(t-t_N)$$

$$= (t^2-1)L_N'(t) \tag{25}$$

Combining (25) with the Legendre derivative relation (16) from Appendix B,

$$w'(t_k) = [(t_k^2 - 1)L'_N(t_k)]' = N(N+1)L_N(t_k) \tag{26}$$

Substituting (25) and (26) into (24) we yield an expression for the Lagrange interpolating functions in terms of the Legendre polynomials.

$$\ell_k(t) = \frac{1}{N(N+1)L_N(t_k)} \frac{(t^2-1)L'_N(t)}{t - t_k}$$

Once we have the Lagrange polynomials in terms of the Legendre polynomials we can analytically compute the weights for LGL quadrature integration,

$$w_k = \int_{-1}^{1} \ell_k(t)dt = \frac{2}{N(N+1)} \frac{1}{[L_N(t_k)]^2}, \quad i = 0, 1, \ldots, N. \tag{27}$$

F Derivative Matrix

$D_{ik} = \dot{\ell}_k(t_i)$ is an $(N+1) \times (N+1)$ matrix. Taking the time derivative of (7), and using the Legendre relation (16) from Appendix B for the derivative of the numerator yields,

$$\left[\frac{\partial}{\partial t}\ell_k(t)\right]_{t=t_i} = \frac{1}{N(N+1)L_N(t_k)} \left[\frac{N(N+1)L_N(t_i)}{t_i - t_k} - \frac{(t_i^2 - 1)L'_N(t_i)}{(t_i - t_k)^2}\right] \tag{28}$$

For any $i \neq k$, the second term in the brackets is zero, since $t = t_i$ is a zero of $(t^2-1)L'_N(t)$. Canceling terms, yields the first component of the derivative matrix in (9). For $i = k$, we utilize l'Hopital's rule for each term (we use the rule twice for the second term).

$$\lim_{t \to t_k} \frac{N(N+1)L_N(t)}{(t - t_k)} = \lim_{t \to t_k} \frac{N(N+1)L'_N(t)}{1} = N(N+1)L'_N(t_k) \tag{29}$$

$$\lim_{t \to t_k} \frac{(t^2-1)L'_N(t)}{(t-t_k)^2} = \lim_{t \to t_k} \frac{N(N+1)L_N(t)}{2(t-t_k)} = \lim_{t \to t_k} \frac{N(N+1)L'_N(t)}{2} = \frac{N(N+1)L'_N(t_k)}{2} \tag{30}$$

For $i = k \neq 0, N$, $L'_N(t_k) = 0$ which indicates that $\dot{\ell}_k(t_k) = 0$ if $k \neq 0, N$. Substituting the values of $L_N(\pm 1)$ and $L'_N(\pm 1)$ given by equations (17-18) in Appendix B and combining these two terms,

$$\frac{\partial \ell_k(t_k)}{\partial t} = (\pm 1)\frac{N(N+1)}{4} \qquad k \in \{0, N\} \tag{31}$$

Therefore, the elements of D are as given in (9),

$$D_{ik} = \begin{cases} \frac{L_N(t_i)}{L_N(t_k)} \frac{1}{t_i - t_k} & i \neq k \\ -\frac{N(N+1)}{4} & i = k = 0 \\ \frac{N(N+1)}{4} & i = k = N \\ 0 & \text{otherwise.} \end{cases}$$

The matrix D is the first differentiation matrix. Optimal control requires only the first derivative, as the differential equation is of order 1. Similar expressions can be computed for second, third, etc. differentiation matrices.

G AMPL Examples

G.1 Single Spin Bloch Optimization

Problem Definition: max $x(T)$, $T = \pi/2$, subject to $u(t)^2 + v(t)^2 \leq 1, t \in [0, T]$ and

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 & 0 & u \\ 0 & 0 & -v \\ -u & v & 0 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix}, \qquad \begin{bmatrix} x(0) \\ y(0) \\ z(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{32}$$

```
param N > 0 integer;
param A > 0;
param T > 0;

param x0;
param y0;
param z0;

set nodes := 1..(N+1);

param D {nodes,nodes};

var x {nodes} >= -1, <= 1;
var y {nodes} >= -1, <= 1;
var z {nodes} >= -1, <= 1;

var u {nodes} >=-A, <=A;
var v {nodes} >=-A, <=A;

maximize cost: x[N+1];

subject to dynamics_x {t in nodes}:
    u[t]*z[t] = (2/T)*(sum{k in nodes} D[t,k]*x[k]);

subject to dynamics_y {t in nodes}:
    -v[t]*z[t] = (2/T)*(sum{k in nodes} D[t,k]*y[k]);

subject to dynamics_z {t in nodes}:
    -u[t]*x[t] + v[t]*y[t] = (2/T)*(sum{k in nodes} D[t,k]*z[k]);

subject to initialConditions_x: x[1] = x0;
subject to initialConditions_y: y[1] = y0;
subject to initialConditions_z: z[1] = z0;

subject to amplitudeBound {t in nodes}: u[t]^2+v[t]^2 <= A^2;

data;
param N := 10;
param A := 1;
param T := pi/2;

param x0 := 0;
param y0 := 0;
param z0 := 1;
param D :
    1    2    3    4    5    6    7    8    9    10    11 :=
    1    -27.50   37.20  -14.88   8.49  -5.64   4.06  -3.06   2.35  -1.79   1.26  -0.50
    2     -6.17   0.00     8.73  -4.07   2.53  -1.77   1.31  -1.00   0.76  -0.53   0.21
    3      1.44  -5.11     0.00   5.25  -2.53   1.61  -1.14   0.85  -0.63   0.44  -0.17
    4      ...
    5      ...
    6      ...
    7      ...
    8      ...
    9      ...
    10     ...
    11     0.50  -1.26    1.79   -2.35   3.06  -4.06   5.64  -8.49  14.88  -37.20  27.50
;
```

G.2 Broadband Spin Bloch Optimization

Problem Definition: max $\int_\Omega x(T,\omega)d\omega$, $0 \le T \le 2\pi$, $\Omega = [-1,1]$ subject to $u(t)^2 + v(t)^2 \le 1, t \in [0,T]$,

$$\frac{d}{dt}\begin{bmatrix} x(t,\omega) \\ y(t,\omega) \\ z(t,\omega) \end{bmatrix} = \begin{bmatrix} 0 & -\omega & u \\ \omega & 0 & -v \\ -u & v & 0 \end{bmatrix} \begin{bmatrix} x(t,\omega) \\ y(t,\omega) \\ z(t,\omega) \end{bmatrix}, \qquad \begin{bmatrix} x(0,\omega) \\ y(0,\omega) \\ z(0,\omega) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{33}$$

```
param N > 0 integer;
param Nw > 0 integer;
param A > 0;
param B > 0;
param Tmax > 0;

param x0;
param y0;
param z0;

set states := 1..3;
set nodes := 1..(N+1);
set dispersion := 1..(Nw+1)

param D {nodes,nodes};
param w {dispersion};
param wwts {dispersion};

var T >= 0 <= Tmax;
var M {states,dispersion,nodes} >= -1, <= 1;

var u {nodes} >=-A, <=A;
var v {nodes} >=-A, <=A;

maximize cost: ((2*B)/2)*(sum{i in dispersion} M[1,i,N+1]*wwts[i]);

subject to dynamics_x {i in dispersion, t in nodes}:
    -w[i]*M[2,i,t] + u[t]*M[3,i,t] = (2/T)*(sum{k in nodes} D[t,k]*M[1,i,k]);

subject to dynamics_y {i in dispersion, t in nodes}:
    w[i]*M[1,i,t] - v[t]*M[3,i,t] = (2/T)*(sum{k in nodes} D[t,k]*M[2,i,k]);
```

```
subject to dynamics_z {i in dispersion, t in nodes}:
    -u[t]*M[1,i,t] + v[t]*M[2,i,t] = (2/T)*(sum{k in nodes} D[t,k]*M[3,i,k]);

subject to initialConditions_x {i in dispersion}:
    M[1,i,1] = x0;
subject to initialConditions_y {i in dispersion}:
    M[2,i,1] = y0;
subject to initialConditions_z {i in dispersion}:
    M[3,i,1] = z0;

subject to amplitudeBound {t in nodes}: u[t]^2+v[t]^2 <= A^2;

data;
param N := 10;
param Nw := 4;
param A := 1;
param B := 1;
param Tmax := 2*pi;

param x0 := 0;
param y0 := 0;
param z0 := 1;

param w :=
  1   -1
  2   -0.5
  3   0
  4   0.5
  5   1
;

param wwts :=
  1   0.1
  2   0.54
  3   0.71
  4   0.54
  5   0.1
;

param D :
    ...
;
```

References

[1] J.-S. Li, J. Ruths, and D. Stefanatos, J. Chem. Phys. 131, 164110 (2009).
PS method applied to quantum systems

[2] G. Elnagar, M. A. Kazemi, and M. Razzaghi, IEEE Trans. Autom. Control 40, 1793 1995.
Short, one of the original papers to use PS methods for optimal control

[3] I. Ross and F. Fahroo, in *New Trends in Nonlinear Dynamics and Control*, edited by W. Kang et al. (Springer, Berlin, 2003); pp. 327.

Good intro to PS methods and discusses "covector mapping theorem" dealing with the order of discretization and transformation to the adjoint system.

[4] F. Fahroo and I. Ross, J. Guid. Control Dynam. 24, 270 2001.
Analyzes the adjoint system for the optimal control problem.

[5] C. Canuto, M. Y. Hussaini, A. Quarteroni and T. A. Zang, *Spectral Methods* (Springer, Berlin, 2006).
Major reference for spectral methods. Very little explicitly about PS methods, but quite a bit of pertinent background material.

[6] G. Szego, *Orthogonal Polynomials* (American Mathematical Society, New York, 1959).
Book on properties of polynomials, Legendre and Lagrange

[7] J. Boyd, *Chebyshev and Fourier Spectral Methods* (Dover Ed. 2, New York, 2000).
Free online pdf book that has a few more direct PS sections than the Canuto book.

[8] B. Fornberg, *A Practical Guide to Pseudospectral Methods*, (Cambridge University Press, New York, 1998).
Book devoted to PS methods and includes several numerical tricks. Approaches it from the Finite-Element perspective.

[9] S. Smith, Ann. Math. Informaticae 33, 109 2006.
Discusses the optimality of the LGL nodes for interpolation

[10] P. Williams, ANZIAM 47, C101 2006.
Application of PS methods and shows some derivation in other ways.

[11] D. Gottlieb, Y. Hussaini, S. Orszag, in *Spectral Methods for Partial Differential Equations*, edited by R. Voigt, D. Gottlieb, and M. Y. Hussaini (SIAM, Philadelphia, 1984); pp 1.
A review of PS methods within a compilation.