

Hessians in Birkhoff-Theoretic Trajectory Optimization

Isaac M. Ross*

Naval Postgraduate School, Monterey, CA 93943

I. Introduction

Deep inside a trajectory optimization algorithm is an iterative scheme given by,

$$\chi^{k+1} = \chi^k + \delta\chi^k, \quad k = 0, 1, \dots \quad (1)$$

where $\chi^k \in \mathbb{R}^n$ is an optimization variable and $\delta\chi^k \in \mathbb{R}^n$ is a search vector. In a line-search strategy [1–3], the search vector is written as $\delta\chi^k = \alpha^k \mathbf{d}^k$, $\alpha^k \in \mathbb{R}$, $\mathbf{d}^k \in \mathbb{R}^n$, where $\alpha^k > 0$ is called the step length and $\mathbf{d}^k$ is called the search direction. In this approach, the search direction $\mathbf{d}^k$ is found first and the step length α^k is computed afterwards. In a trust region strategy [1–3], the search vector is found by constraining it to a “trust” region $\|\delta\chi^k\| \leq \Delta^k > 0$ where $\|\cdot\|$ is typically a 2-norm and Δ^k is the trust-region radius. This approach can be contrasted to the line-search strategy as finding the “step length” Δ^k first and computing the search direction $\delta\chi^k$ afterwards. In either case, the computation of $\delta\chi^k$ involves solving a linear matrix equation,

$$\mathbf{A}^k \boldsymbol{\xi}^k = \mathbf{g}^k, \quad k = 0, 1, \dots \quad (2)$$

where $\mathbf{A}^k$ is an $n \times n$ matrix that contains an amalgamation of the discretized Jacobians and Hessians associated with the trajectory optimization problem, $\mathbf{g}^k \in \mathbb{R}^n$ comprises function and Jacobian evaluations based on the problem data functions and $\boldsymbol{\xi}^k$ is either $\mathbf{d}^k$, as in the case of a line-search strategy or $\delta\chi^k$ if the algorithm employs a trust-region. The specifics of the process involved in generating Eq. (2) from Eq. (1) is described in detail in Section IV.

Solving Eq. (2) for $\boldsymbol{\xi}^k$ is one of the most computationally expensive components of a trajectory optimization algorithm. It can be roughly stated as having an asymptotic worst-case computational complexity [4–6] of $\mathcal{O}(n^3)$. For example, Gauss elimination requires $(2/3)n^3$ floating point operations [4, 6]. The $\mathcal{O}(n^3)$ -statement is a more precise version of the well-known adage that if n is small/large the computational time for solving Eq. (2) is correspondingly small/large. Furthermore, because Eq. (2) must be solved for every $k = 0, 1, \dots$, the worst-case computational time is additive and can be expressed as $\mathcal{O}(Kn^3)$, where K is the number of iterations. If $\mathbf{A}^k$ is structured, the computational complexity for solving Eq. (2) can be reduced by employing non-generic methods and algorithms for solving linear matrix equations [6]. A trivial example of the last statement is the case when $\mathbf{A}^k$ is diagonal. In this case, the computational complexity of solving Eq. (2) is just $\mathcal{O}(n)$. To briefly put the arguments of the preceding statements in perspective, consider a value of $n = 1,000$, a small number for a present-day computer. The computational time for $\mathcal{O}(n^3)$ operations is proportional to 10^9 time units. In contrast, an $\mathcal{O}(n)$ method takes a mere 10^3 time units. Stated differently, the $\mathcal{O}(n)$ -computation is a *million-times faster* than $\mathcal{O}(n^3)$ (for $n = 1,000$). In

*Distinguished Professor and Program Director, Control and Optimization, Department of Mechanical and Aerospace Engineering. imross@nps.edu

view of all these considerations, it is apparent that understanding the detailed structure of $\mathbf{A}^k$ is an important problem for generating fast, stable and accurate solutions to trajectory optimization problems.

In this note, we derive the matrix $\mathbf{A}^k$ for the universal Birkhoff-theoretic method for trajectory optimization developed in [7–9]. The contributions of this note are as follows:

1. We use an unconventional weighted Lagrangian and a special scaling of the optimization variables used in [8] to derive $\mathbf{A}^k$. We show that $\mathbf{A}^k$ can be split into two blocks: one that contains only the Birkhoff-specific components and another that depends only on the problem-specific data functions. That is, the Birkhoff-specific components are decoupled from the problem-specific elements.
2. We show that the problem-specific components of $\mathbf{A}^k$ are block diagonal matrices that can be expressed explicitly and only pointwise in terms of the data functions (and their derivatives).
3. We show that a particular arrangement of alternating primal and dual variables together with a particular listing of the constraints generates a matrix $\mathbf{A}^k$ whose diagonal entries are mostly identity matrices.
4. Besides expressing it in an interesting manner, we show that the off-diagonal blocks of $\mathbf{A}^k$ can be directly connected, via an identity transform, to the Hessian of the Hamiltonian, the Hessian of the endpoint Lagrangian and the Jacobian of the problem data functions.
5. To demonstrate the possible utilization of the special structure of $\mathbf{A}^k$, we develop an eigenvalue theorem for the Birkhoff Hessian based on the well-known Gershgorin theorem [10]. Key results of this new theorem are:
 - (a) The spectral radius of the Birkhoff Hessian is mesh-independent;
 - (b) Approximately, 80% of the eigenvalues are independent of the problem data; and,
 - (c) The real components of the problem-independent eigenvalues (i.e., approximately 80% of the eigenvalues) are all contained in the narrow interval $[-2, 4]$.
6. We use the preceding results to quantify the computational complexity [11] of solving Eq. (2) and point a way forward on how to address the grand challenge of solving a million-point trajectory optimization problem [12].

The results of this note also suggest that it is possible to make informed changes to the trajectory optimization problem (such as coordinate transformations, scaling etc. [13]) with immediate and detailed knowledge of their impact on the innermost component of an algorithm, namely Eq. (1) and Eq. (2).

II. An Overview of Birkhoff-Theoretic Discretizations of Optimal Control Problems

As illustrated in Fig. 1, a universal Birkhoff-theoretic method for trajectory optimization [8, 9, 14] generates an extremely large number of variants. To limit the scope of this note, we consider only those Birkhoff methods that use orthogonal polynomials as basis functions. In particular, we limit the scope of this note to the Birkhoff theory that utilizes only the Chebyshev and Legendre

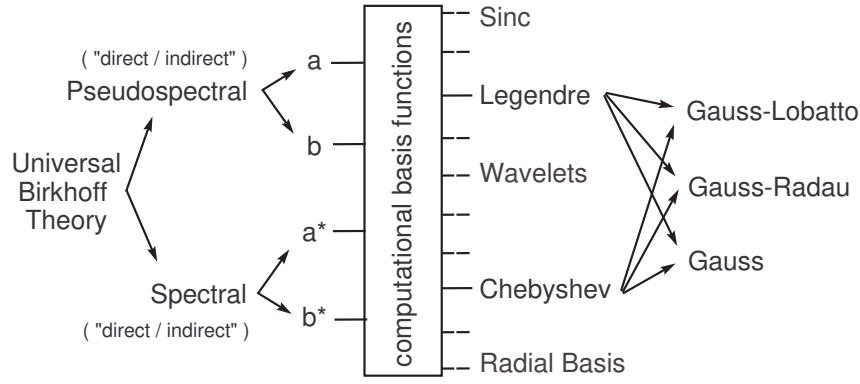


Figure 1. Schematic for illustrating the plethora of possible Birkhoff-theoretic methods for trajectory optimization; figure adapted from [9] and [14].

polynomial basis functions [7–9]. Because of the choice of these Gegenbauer (i.e., Chebyshev and Legendre) basis functions, we further limit the scope of this note by constraining the independent variable to the domain $[-1, 1] = [\tau^a, \tau^b]$ in order to circumvent the impact of time domain transformation techniques [15–17]. Finally, as has been noted many times before [8, 16, 18, 19], it is more informative to explain the key ideas by formulating a distilled optimal control problem that avoids the distractions of bookkeeping the multitude of variables and constraints. Because the main point of this note is to illuminate the structure of the Hessian, we define a particular version of a distilled optimal control problem as follows:

$$(P) \begin{cases} \text{Minimize} & J[x(\cdot), u(\cdot)] := E(x(\tau^a), x(\tau^b)) \\ \text{Subject to} & \dot{x}(\tau) = f(x(\tau), u(\tau)) \\ & e(x(\tau^a), x(\tau^b)) = 0 \end{cases} \quad (3)$$

The data in Problem (P), as indicated in Eq. (3), are as follows: J is a cost functional given by a twice-differentiable endpoint (“Mayer” [15]) cost function $E : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$; the pair, $(x(\cdot), u(\cdot))$, is the unknown system trajectory (i.e., state-control function pair, $\tau \mapsto (x(\tau), u(\tau)) \in \mathbb{R} \times \mathbb{R}$; $f : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ is a given twice-differentiable dynamics function; and, $e : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ is a given twice-differentiable endpoint constraint function that constrains the endpoint values of $x(\tau)$ at $\tau^a = -1$ and $\tau^b = 1$.

A universal Birkhoff method [7, 8] begins by choosing a grid $\pi^N := \{\tau_0, \dots, \tau_N\}$, $\tau^a \leq \tau_0 < \dots < \tau_N \leq \tau^b$ that allows $\tau \mapsto x(\tau)$ to be approximated by two equivalent a - and b -expansions:

$$x^N(\tau) := x^a B_0^0(\tau) + \sum_{j=0}^N v_j B_j^a(\tau) \quad (4a)$$

$$x^N(\tau) := \sum_{j=0}^N v_j B_j^b(\tau) + x^b B_N^N(\tau) \quad (4b)$$

where $(x^a, x^b) = (x(\tau^a), x(\tau^b))$, $B_0^0(\tau)$, $B_N^N(\tau)$, $B_j^a(\tau)$ and $B_j^b(\tau)$, $j = 0, \dots, N$ are the Birkhoff

basis functions that satisfy the following interpolating conditions [7,8]:

$$\begin{aligned}
B_0^0(\tau^a) &= 1, & B_N^N(\tau^b) &= 1, \\
d_\tau B_0^0(\tau_i) &= 0, & d_\tau B_N^N(\tau_i) &= 0, & i &= 0, \dots, N \\
B_j^a(\tau^a) &= 0, & B_j^b(\tau^b) &= 0, & j &= 0, \dots, N \\
d_\tau B_j^a(\tau_i) &= \delta_{ij}, & d_\tau B_j^b(\tau_i) &= \delta_{ij}, & i &= 0, \dots, N, j = 0, \dots, N
\end{aligned} \tag{5}$$

In Eq. (5), d_τ is the derivative operator, $d/d\tau$, and δ_{ij} is the Kronecker delta. The variables $v_j, j = 0, \dots, N$ in Eq. (4) are called the virtual variables or the virtual control variables [8,14].

Let, $x^N(\tau_i) = x_i$ and $u(\tau_i) = u_i, i = 0, \dots, N$. Define [7,8],

$$\begin{aligned}
X &:= [x_0, \dots, x_N]^T \\
V &:= [v_0, \dots, v_N]^T \\
U &:= [u_0, \dots, u_N]^T \\
f(X, U) &:= [f(x_0, u_0), \dots, f(x_N, u_N)]^T \\
\mathbf{w}_B &:= [w_0, \dots, w_N]^T \\
\mathbf{b} &:= [1, \dots, 1]^T \\
[\mathbf{B}^a]_{ij} &:= [B_j^a(\tau_i)] \\
[\mathbf{B}^b]_{ij} &:= [B_j^b(\tau_i)] \\
N_n &:= N + 1
\end{aligned} \tag{6}$$

where $w_i, i = 0, \dots, N$ are the Birkhoff quadrature weights [7]. Then, the a -version of the Birkhoff discretization of Problem (P) is given by [8],

$$(P_a^N) \left\{ \begin{array}{ll} \text{Minimize} & J_a^N[X, U, V, x^a, x^b] := E(x^a, x^b) \\ \text{Subject to} & X = x^a \mathbf{b} + \mathbf{B}^a V \\ & V = f(X, U) \\ & x^b = x^a + \mathbf{w}_B^T V \\ & e(x^a, x^b) = 0 \end{array} \right. \tag{7}$$

Problem (P_a^N) is a discretized optimal control problem (DOCP). The discretization refers to time while the optimization variables X, V, U, x^a and x^b are all continuous. In much of the trajectory optimization literature [20,21], DOCPs (such as Problem (P_a^N)) are called nonlinear programming problems (NLPs). In principle, this is true; however, note that the variables X, V, U, x^a and x^b have different and special properties. In an NLP these variables are treated as a single parameter, $\mathbf{y} := (X, V, U, x^a, x^b)$ that is largely agnostic to any special properties of its constituent variables vis-a-vis its continuous-time counterparts. As noted elsewhere [14,22,23], DOCPs occupy a much smaller “space” than the “space of all NLPs.” In view of this, we approach a DOCP as a very special continuous optimization problem that is linked to Problem (P) in every step of the resulting analysis up to and including a connection to Eq. (1) and Eq. (2). It will be apparent in the sections to follow that this unique treatment of Problem (P_a^N) is sharply different from the conventional “NLP blackbox” perspective.

III. Connecting a Special Lagrangian to the Pontryagin Hamiltonian

To draw a connection between a Lagrangian of a DOCP and the Pontryagin Hamiltonian of Problem (P), we need the following definition and lemma (proved in [8]):

Definition 1 *The Birkhoff quadrature matrix is the following $N_n \times N_n$ matrix,*

$$W_B := \text{diag}[\mathbf{w}_B] := \begin{bmatrix} w_0 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & w_N \end{bmatrix} \quad (8)$$

where $\mathbf{w}_B$ is as given in Eq. (6).

Lemma 1 ([8]) *Given any $\epsilon > 0$, there exists an $N_\epsilon \in \mathbb{N}$ such that for all $N \geq N_\epsilon$ the following holds:*

$$W_B \mathbf{B}^b + [\mathbf{B}^a]^T W_B = \mathbf{0}$$

Based on results of Theorem 3 of [8], we scale the variables X, V and U in the DOCP given by Problem (P_a^N) by W_B such that the new optimization variables $\tilde{X}, \tilde{V}$ and $\tilde{U}$ are given by,

$$\begin{aligned} \tilde{X} &= \mathbf{w}_B \circ X = W_B X \\ \tilde{V} &= \mathbf{w}_B \circ V = W_B V \\ \tilde{U} &= \mathbf{w}_B \circ U = W_B U \end{aligned}$$

where $\circ$ denotes a Hadamard operation. In not treating a DOCP as a generic NLP, consider the following specially weighted Lagrangian for Problem (P_a^N):

$$\begin{aligned} L(\Omega, \Lambda, \lambda^b, \nu, \tilde{X}, \tilde{U}, \tilde{V}, x^a, x^b) &:= E(x^a, x^b) \\ &+ \Omega^T W_B \left(W_B^{-1} \tilde{X} - \mathbf{B}^a W_B^{-1} \tilde{V} - x^a \mathbf{b} \right) \\ &+ \Lambda^T W_B \left(f(W_B^{-1} \tilde{X}, W_B^{-1} \tilde{U}) - W_B^{-1} \tilde{V} \right) \\ &+ \lambda^b \left(x^a + \mathbf{w}_B^T W_B^{-1} \tilde{V} - x^b \right) + \nu e(x^a, x^b) \end{aligned} \quad (9)$$

where $\Omega, \Lambda, \lambda^b$ and ν are the Lagrange multipliers associated with the constraints implied in Eq. (9). Under these constructs, the necessary conditions for the variable-scaled DOCP are given by,

$$\partial_{\tilde{X}} L = 0 = \partial_{\tilde{V}} L = \partial_{\tilde{U}} L = \partial_{x^a} L = \partial_{x^b} L \quad (10)$$

To apply Eq. (10) in an efficient manner, it is useful to note that

$$\begin{aligned} \partial_X f(X, U) &= \begin{bmatrix} \partial_x f(x_0, u_0) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \partial_x f(x_N, u_N) \end{bmatrix} \\ &:= \text{diag}[\partial_x f(x_0, u_0), \dots, \partial_x f(x_N, u_N)]^T \end{aligned}$$

Defining [8],

$$\partial_x f(X, U) := [\partial_x f(x_0, u_0), \dots, \partial_x f(x_N, u_N)]^T \quad (11)$$

it follows that we can write,

$$\partial_X f(X, U) = \text{diag}[\partial_x f(X, U)]$$

Hence we have,

$$[\partial_X f(X, U)]^T \Lambda = \text{diag}[\partial_x f(X, U)] \Lambda = \partial_x f(X, U) \circ \Lambda$$

With these notational conveniences, an application of Eq. (10) using Eq. (9) and Lemma 1 generates the following set of equations,

$$\Omega + \partial_x f(X, U) \circ \Lambda = 0 \quad (12a)$$

$$\Lambda - \lambda^b \mathbf{b} - \mathbf{B}^b \Omega = 0 \quad (12b)$$

$$\partial_u f(X, U) \circ \Lambda = 0 \quad (12c)$$

$$\lambda^a + \partial_{x^a} \bar{E}(\nu, x^a, x^b) = 0 \quad (12d)$$

$$\lambda^b - \partial_{x^b} \bar{E}(\nu, x^a, x^b) = 0 \quad (12e)$$

where $\partial_u f(X, U)$ is defined similarly to Eq. (11), λ^a is defined by [8],

$$\lambda^a := \lambda^b - \mathbf{w}_B^T \Omega \quad (13)$$

and $\bar{E}(\nu, x^a, x^b)$ is the endpoint Lagrangian [15] function (associated with Problem (P)) defined by,

$$\bar{E}(\nu, x^a, x^b) := E(x^a, x^b) + \nu e(x^a, x^b) \quad (14)$$

The purpose of the preceding mathematical machinations is now clear, at least, partially. Observe that Eq. (12d) and Eq. (12e) are, in fact, the transversality conditions [15, 24] for Problem (P_a). In fact, the remainder of the equations in Eq. (12) are discretizations of the remainder of the first-order necessary conditions of Pontryagin [15]. To make the preceding statement more precise, let $H(\lambda, x, u)$ denote the Pontryagin Hamiltonian [15] for Problem (P) given by,

$$H(\lambda, x, u) := \lambda f(x, u) \quad (15)$$

The necessary condition for Hamiltonian minimization is given by,

$$\partial_u H(\lambda(\tau), x(\tau), u(\tau)) = \lambda(\tau) \partial_u f(x(\tau), u(\tau)) = 0 \quad \forall \tau \quad (16)$$

Hence, over the grid π^N , Eq. (16) maps to,

$$\partial_u H(\lambda(\tau_i), x(\tau_i), u(\tau_i)) = \lambda(\tau_i) \partial_u f(x(\tau_i), u(\tau_i)) = 0 \quad \forall \tau_i, i = 0, \dots, N \quad (17)$$

Using the notational convention defined in Eq. (11), it follows that Eq. (17) can be written as,

$$\partial_u H(\Lambda, X, U) = \Lambda \circ \partial_u f(X, U) = 0 \quad (18)$$

Hence, Eq. (12c) is identical to the Hamiltonian form given by Eq. (18). Similarly, it is straightforward to show that

$$\partial_x H(\Lambda, X, U) = \Lambda \circ \partial_x f(X, U) \quad (19)$$

Using Eq. (19) it can be shown [8] that Eq. (12a) and Eq. (12b) are the Birkhoff discretizations of the adjoint equation,

$$-\dot{\lambda}(\tau) = \partial_x H(\lambda, x, u) \quad (20)$$

written in terms of the differential-algebraic equation,

$$\dot{\lambda}(\tau) := \omega(\tau), \quad \omega(\tau) + \partial_x H(\lambda, x, u) = 0 \quad (21)$$

where $\omega(\tau)$ is called the co-virtual variable [8, 14]. Collecting all relevant equations, it follows that Eq. (12) and hence Eq. (10) can be directly connected to Problem (P) in the sense that it can be rewritten in terms of the discretized Pontryagin Hamiltonian as follows:

$$\Omega + \partial_x H(\Lambda, X, U) = 0 \quad (22a)$$

$$\Lambda - \lambda^b \mathbf{b} - \mathbf{B}^b \Omega = 0 \quad (22b)$$

$$\lambda^a - \lambda^b + \mathbf{w}_B^T \Omega = 0 \quad (22c)$$

$$\partial_u H(\Lambda, X, U) = 0 \quad (22d)$$

$$\lambda^a + \partial_{x^a} \bar{E}(\nu, x^a, x^b) = 0 \quad (22e)$$

$$\lambda^b - \partial_{x^b} \bar{E}(\nu, x^a, x^b) = 0 \quad (22f)$$

Remark 1 When a generic unweighted Lagrangian is differentiated with respect to its multipliers, it regenerates the equality constraints of an NLP [1–3]. It is straightforward to show that differentiating the special Lagrangian of Eq. (9) with respect to Ω and Λ generates the constraints in Problem (P_a^N) in its spectral form [8]. That is, the resulting constraints are multiplied by W_B . This is a weak representation of the constraints because for $N > 2$, $0 < w_i < 1, i = 0, \dots, N$. To impose the primal constraints strongly, we use its formulation in its original “strong” (pseudospectral) form given in Eq. (7).

Remark 2 Equation (13) is an “artifice” that does not naturally occur if Problem (P_a^N) is treated as a generic NLP even if the special Lagrangian of Eq. (9) is used. In the context of an NLP, Eq. (13) is simply a clever grouping of the NLP multipliers to produce a “new” linear equation. In the context of a trajectory optimization problem, Eq. (13) is a critical equation in the sense that λ^a is the initial costate.

Remarks 1 and 2 illustrate why it may be unwise to treat Problem (P_a^N) a generic NLP. In fact, the following sections amplify this point.

IV. Hamiltonian Form of the Birkhoff Hessian

In connecting the ideas introduced in Section I to Section III, let χ tentatively denote the arguments of the special Lagrangian given by Eq. (9). Then, from the results of Section III, it follows that a solution to Problem (P_a^N) can be obtained by solving a system of linear and nonlinear equations that can be expressed as $\mathcal{F}(\chi) = 0$, the details of which will be presented shortly. Newton’s method for solving a system of nonlinear equations is given by [1–3],

$$\partial_{\chi} \mathcal{F}(\chi^k) d^k = -\mathcal{F}(\chi^k) \quad (23)$$

where $d^k := \chi^{k+1} - \chi^k$ is the Newton step. See also Eq. (1) and Eq. (2) for context. Because the source of $\mathcal{F}(\chi)$ is the gradient of the Lagrangian (see Eq. (12)), the Jacobian in Eq. (23) is the Hessian of $L(\Omega, \Lambda, \lambda^b, \nu, \tilde{X}, \tilde{U}, \tilde{V}, x^a, x^b)$, under the caveat of Remark 1. Hence, we refer to such a Jacobian as a Birkhoff Hessian. The task at hand is to develop a Birkhoff Hessian that can be directly connected to Problem (P).

To construct a Birkhoff Hessian that is revelatory with respect to its properties, we first organize all the variables in a particular arrangement of mostly alternating primal and dual variables according to (see also Remark 2),

$$\begin{aligned}\chi &:= (X, \Lambda, V, \Omega, U, x^a, \lambda^b, x^b, \nu, \lambda^a) \\ &\in \mathbb{R}^{N_n} \times \mathbb{R}^{N_n} \times \mathbb{R}^{N_n} \times \mathbb{R}^{N_n} \times \mathbb{R}^{N_n} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}\end{aligned}\quad (24)$$

For similar reasons, we also arrange the components of $\mathcal{F}(\chi)$ (i.e., Eq. (22) and the constraints in Eq. (7)) in the following specific manner:

$$\mathcal{F}_1(\chi) = X - x^a \mathbf{b} - \mathbf{B}^a V \quad (25a)$$

$$\mathcal{F}_2(\chi) = \Lambda - \lambda^b \mathbf{b} - \mathbf{B}^b \Omega \quad (25b)$$

$$\mathcal{F}_3(\chi) = V - \partial_\lambda H(\Lambda, X, U) \quad (25c)$$

$$\mathcal{F}_4(\chi) = \Omega + \partial_x H(\Lambda, X, U) \quad (25d)$$

$$\mathcal{F}_5(\chi) = \partial_u H(\Lambda, X, U) \quad (25e)$$

$$\mathcal{F}_6(\chi) = x^a - x^b + \mathbf{w}_B^T V \quad (25f)$$

$$\mathcal{F}_7(\chi) = -\lambda^a + \lambda^b - \mathbf{w}_B^T \Omega \quad (25g)$$

$$\mathcal{F}_8(\chi) = -\lambda^b + \partial_{x^b} \overline{E}(\nu, x^a, x^b) \quad (25h)$$

$$\mathcal{F}_9(\chi) = \partial_\nu \overline{E}(\nu, x^a, x^b) \quad (25i)$$

$$\mathcal{F}_{10}(\chi) = \lambda^a + \partial_{x^a} \overline{E}(\nu, x^a, x^b) \quad (25j)$$

Note the sign changes in $\mathcal{F}_7(\chi)$ and $\mathcal{F}_8(\chi)$. Note also that $\mathcal{F}_3(\chi)$ and $\mathcal{F}_9(\chi)$ are the primal constraint equations (i.e., the constraints in Eq. (7)) rewritten in terms of the Hamiltonian and the endpoint Lagrangian respectively using the fact that,

$$\partial_\lambda H(\lambda, x, u) = f(x, u), \quad \partial_\nu \overline{E}(\nu, x^a, x^b) = e(x^a, x^b) \quad (26)$$

Hence (see Eq. (11)),

$$\begin{aligned}\partial_\lambda H(\Lambda, X, U) &= f(X, U) \\ &:= [f(x_0, u_0), \dots, f(x_N, u_N)]^T \\ &= \partial_\Lambda H(\Lambda, X, U)\end{aligned}$$

Furthermore,

$$\partial_{\Lambda, X}^2 H(\Lambda, X, U) = \partial_X f(X, U) = \begin{bmatrix} \partial_x f(x_0, u_0) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \partial_x f(x_N, u_N) \end{bmatrix} \quad (27)$$

Similarly, it is straightforward to show that the second derivative of the Hamiltonian with respect to any of its variables (Λ, X, U) is a diagonal matrix.

To simplify the identification of the matrix subblocks in the construction of the Birkhoff Hessian, we use the following notational convenience:

$$\partial_{\Lambda, X}^2 H(\Lambda, X, U) \equiv H_{\Lambda, X}, \quad \partial_{x^a, x^b}^2 \overline{E} \equiv \overline{E}_{a, b}$$

With these conveniences, the Birkhoff Hessian associated with the DOCP given by Problem (P_a^N) (i.e., Jacobian of $\mathcal{F}(\chi)$) can now be written succinctly as,

$$\mathbf{A} := \begin{bmatrix} I & 0 & -\mathbf{B}^a & 0 & 0 & -\mathbf{b} & 0 & 0 & 0 & 0 \\ 0 & I & 0 & -\mathbf{B}^b & 0 & 0 & -\mathbf{b} & 0 & 0 & 0 \\ -H_{\Lambda,X} & 0 & I & 0 & -H_{\Lambda,U} & 0 & 0 & 0 & 0 & 0 \\ H_{X,X} & H_{X,\Lambda} & 0 & I & H_{X,U} & 0 & 0 & 0 & 0 & 0 \\ H_{U,X} & H_{U,\Lambda} & 0 & 0 & H_{U,U} & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & \mathbf{w}_B^T & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -\mathbf{w}_B^T & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & \bar{E}_{b,a} & -1 & \bar{E}_{b,b} & \bar{E}_{b,\nu} & 0 \\ 0 & 0 & 0 & 0 & 0 & \bar{E}_{\nu,a} & 0 & \bar{E}_{\nu,b} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \bar{E}_{a,a} & 0 & \bar{E}_{a,b} & \bar{E}_{a,\nu} & 1 \end{bmatrix} \quad (28)$$

Some comments about $\mathbf{A}$ are in order:

1. I is an $N_n \times N_n$ identity matrix. The 0's in Eq. (28) are matrices of order $N_n \times N_n$ or $N_n \times 1$ or or $1 \times N_n$ or 1×1 . The precise size of these 0's are apparent from the context.
2. The Hessian is partitioned as indicated in Eq. (28) to indicate that the top left “11-block” corresponds to continuous variables that have values over the entire grid while the bottom right “22-block” corresponds to continuous variables that take values only over the boundary points a and b .
3. When compared to the 22-block, the 11-block is relatively large. The 22-block is 5×5 while the 11-block is $5N_n \times 5N_n$. In other words, the 11-block overwhelmingly dominates in size.
4. The variables and functions (i.e., χ and $\mathcal{F}$) are arranged to show that the identity matrix occupies all the block diagonal elements of the Hessian in the top left block except for the lower right corner. Similarly the diagonal elements of the bottom right block are 0's and 1's except for the the middle 33-element of this sub-block.
5. The first two “rows” (block-wise) of the Hessian matrix, along with “rows” 6 and 7 are problem-independent. All other rows depend only the problem data functions and are decoupled from any specificity with respect to a Birkhoff-theoretic method for trajectory optimization. Except for the bottom right block of the Hessian, all problem-dependent data are located only along the diagonals of the block elements indicated in Eq. (28).

If Problem (P_a^N) were treated as a generic NLP, its Hessian would not be the same as Eq. (28). This is because of the following reasons:

1. In a typical NLP, the special Lagrangian given by Eq. (9) is not used. The use of such a Lagrangian was critical to the development of Eq. (28).
2. Even if an NLP were to be modified to incorporate the special Lagrangian of Eq. (9), Eq. (13) would not appear as part of its multiplier set. In this case the Hessian would be a “ 9×9 ” matrix instead of the “ 10×10 ” form presented in Eq. (28). See Section VII for additional details.

3. Because an NLP organizes variables differently than Eq. (24), the identity matrices indicated in Eq. (28) would lose their prime location along the diagonals. See Section VII for additional details.
4. The Hessian in an NLP is symmetric [1–3]. It is obvious that the Birkhoff Hessian as expressed in Eq. (28) is not symmetric.

Remark 3 *It is straightforward but tedious to show that no rearrangement of the rows/columns of $\mathbf{A}$ will generate a symmetric matrix. A clue for this “theorem of impossibility” is the last column of $\mathbf{A}$. The origin of this column (and $\mathcal{F}_7(\chi)$) is the treatment of Eq. (13) as a constraint.*

The location of the identity matrices along the diagonal can be used judiciously in several different ways. Complete details of utilizing this property for solving an optimal control problem are well beyond the scope of this note. Nonetheless, we prove in the next section a general theorem pertaining to the eigenvalues of $\mathbf{A}$.

V. A Birkhoff Hessian Eigenvalue Theorem

The eigenvalues of a matrix are a critically important component of its fundamental properties [10]. For a non-numerical matrix such as the one given by Eq. (28), it is typically fruitless to find its eigenvalues in terms of its constituent elements. However, Gershgorin’s theorem [10] provides a simple and elegant summation formula to estimate the range of possible eigenvalues of a square matrix. In this section, we prove an eigenvalue theorem by combining (a) Gershgorin’s theorem, (b) the properties of the Birkhoff matrices $\mathbf{B}^a$ and $\mathbf{B}^b$ [7, 8] and (c) the structure of the Hessian as presented in Eq. (28).

Theorem 1 *Let*

1. π^N be the Legendre/Chebyshev (“Gegenbauer”) family of grids indicated in Fig. 1;
2. Problem (P), given by Eq. (3), be discretized over π^N by a universal Birkhoff-theoretic method; and
3. $\mathbf{A}$, given by Eq. (28), be the Hessian matrix associated with a Birkhoff-Gegenbauer discretization of Problem (P).

Then, the spectral radius of the Birkhoff Hessian is mesh/grid-independent; i.e., independent of N . Furthermore, the real part(s) of:

1. $4N_n + 2$ eigenvalues are numerically bounded by the interval $[-2, 4]$, ($N_n = N + 1$).
2. One eigenvalue is bounded by the interval $[0, 2]$.
3. One eigenvalue is bounded by $\mp (|\overline{E}_{\nu,a}| + |\overline{E}_{\nu,b}|)$
4. One eigenvalue is bounded by $|\overline{E}_{b,b}| \mp R^E$, where $R^E := 1 + |\overline{E}_{b,a}| + |\overline{E}_{b,\nu}|$.
5. N_n eigenvalues are bounded by $|H_{u_i,u_i}| \mp R_i^H$, $i = 0, \dots, N$, where $R_i^H := |H_{u_i,x_i}| + |H_{u_i,\lambda_i}|$

Proof. We will first prove Statements 1–5. Then, the assertion that the spectral radius of the Hessian is independent of N will be obvious.

We prove Statement 1 in three parts:

1. First, consider the identity matrices occupying the 11- and 22-blocks of $\mathbf{A}$. For these blocks, consider applying Gershgorin's theorem row wise. Each element of the vector $\mathbf{b}$ is unity. Hence, the absolute value of any row of $-\mathbf{b}$ is unity. The sum of the absolute values of any row of $\mathbf{B}^a$ or $\mathbf{B}^b$ is less than or equal to 2 [7]. Hence the sum of the absolute values of the off diagonal elements of the first two rows of $\mathbf{A}$ is 3. Applying Gershgorin's theorem, it follows that the real parts of $2N_n$ eigenvalues are centered at one with a radius of 3. Or equivalently, the real parts of $2N_n$ eigenvalues lie in the interval $[1 - 3, 1 + 3] = [-2, 4]$.
2. Next, consider the identity matrices occupying the 33- and 44-blocks of $\mathbf{A}$. For these blocks, we apply Gershgorin's theorem column wise. The absolute value of each element of $\mathbf{w}_B$ is at most unity [7]. Hence, it follows (from the results of the previous paragraph) that the real parts of an additional $2N_n$ eigenvalues lie in the interval $[-2, 4]$ raising the tally to $4N_n$.
3. Finally, consider the 1's occupying the 66- and 77 "blocks" of $\mathbf{A}$. Applying Gershgorin's theorem row wise to these blocks results in (see previous paragraph) two more eigenvalues lying in the interval $[-2, 4]$.

Thus, Statement 1 is proved. To prove Statement 2 of the theorem, consider the last column of $\mathbf{A}$. It is obvious that one eigenvalue is between 0 and 2.

To prove Statements 3, 4 and 5, we apply Gershgorin's theorem (row or column wise) to the diagonal entries of $\mathbf{A}$ occupying the 99-, 88- and 55-blocks of $\mathbf{A}$ respectively.

From the proofs of Statements 1–5, it follows that the spectral radius of the Hessian is independent of N . $\square$

Remark 4 For the 77-block of $\mathbf{A}$, applying Gershgorin's theorem along the column generate an estimate of an eigenvalue that is dependent upon N . Choosing row 7 in the proof of Theorem 1 was critical to proving mesh independence.

Remark 5 Imbedded in Eq. (28) are at least six different possible Hessians based on the choice of the Gegenbauer grid (see Fig. 1). Hence Theorem 1 imbeds at least six grid-dependent theorems.

VI. A Preliminary Analysis of Computational Complexity

The computational cost of solving solving Eq. (2) with $\mathbf{A}$ given by Eq. (28) can be classified in terms of space and time complexity [11]. To facilitate a simplified and practical discussion on this topic, a collection of space and time complexity numbers are provided in Table 1 in a manner that is relatable to a present-day processor. The computational space is computer memory listed in

variables	Memory (GB)		Compute Time (s/TFLOPS)			
n	n	n^2	n	$n \log(n)$	n^2	n^3
1,000	0.000008	0.008	1×10^{-9}	7×10^{-9}	0.000001	0.001
10,000	0.00008	0.8	1×10^{-8}	9×10^{-8}	0.0001	1
100,000	0.0008	80	1×10^{-7}	11×10^{-7}	0.01	1,000
1,000,000	0.008	8,000	1×10^{-6}	14×10^{-6}	1	1,000,000

Table 1. Space and time complexity table baselined to gigabytes and terraflops

gigabytes (GB) in Table 1 where n is a generic number of variables. Its relationship to the number

of variables in trajectory optimization will be connected shortly. The computational time in Table 1 is in seconds, baselined to a terraflops (TFLOPS) processor. An example of how to read this table is as follows: Suppose we have a 1,000-variable problem. Storing a square matrix for $n = 1,000$ requires 0.008 GB of memory ($= n^2$ in Table 1, namely, row 1, column 3). If Eq. (2) is solved using an algorithm that has $\mathcal{O}(n^3)$ complexity, then it can be solved in about 0.001 seconds (row 1, column 7). We caveat this number and all of the discussions to follow with the fact that actual compute time depends on many more factors than FLOPS such as the number of pipelines, pipeline length, cache size, cache latency, instruction sets, number of registers, etc. [25]. Consequently, we use the compute times provided in Table 1 to simply provide a relatable number in seconds. See also [8] for a critique on using actual computer time for solving a problem to indicate “fast” in fast trajectory optimization.

As noted in Section I, solving Eq. (2) is one of the most computationally expensive components of a trajectory optimization algorithm. To analyze this computational cost, we permute the rows of $\mathbf{A}$ given by Eq. (28) to cast it in the following form:

$$P\mathbf{A} := \begin{bmatrix} \mathbf{A}_0 \\ \mathbf{A}_{data} \end{bmatrix} := \begin{bmatrix} I & 0 & -\mathbf{B}^a & 0 & 0 & -\mathbf{b} & 0 & 0 & 0 & 0 \\ 0 & I & 0 & -\mathbf{B}^b & 0 & 0 & -\mathbf{b} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{w}_B^T & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -\mathbf{w}_B^T & 0 & 0 & 1 & 0 & 0 & -1 \\ \hline -H_{\Lambda,X} & 0 & I & 0 & -H_{\Lambda,U} & 0 & 0 & 0 & 0 & 0 \\ H_{X,X} & H_{X,\Lambda} & 0 & I & H_{X,U} & 0 & 0 & 0 & 0 & 0 \\ H_{U,X} & H_{U,\Lambda} & 0 & 0 & H_{U,U} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \bar{E}_{b,a} & -1 & \bar{E}_{b,b} & \bar{E}_{b,\nu} & 0 \\ 0 & 0 & 0 & 0 & 0 & \bar{E}_{\nu,a} & 0 & \bar{E}_{\nu,b} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \bar{E}_{a,a} & 0 & \bar{E}_{a,b} & \bar{E}_{a,\nu} & 1 \end{bmatrix} \quad (29)$$

In Eq. (29), P is a permutation matrix that rearranges $\mathbf{A}$ such that the “top” matrix $\mathbf{A}_0$ is independent of the problem data and $\mathbf{A}_{data}$ is the data-dependent “bottom” matrix.

A. Computational Cost Associated with the Data-Dependent Matrix, $\mathbf{A}_{data}$

From Eq. (27) it follows that the Hessian of the Hamiltonian is a diagonal matrix (in its Birkhoff-discretized form). Ignoring the constants (i.e., both ones and zeros) the sparsity pattern of $\mathbf{A}_{data}$ is shown in Fig. 2. It is apparent by inspection of this figure that space complexity (i.e. computer memory) is not an issue. In terms of specifics, taking the symmetry of the Hessian of the Hamiltonian into account, the required memory for storing the Hessian is only $5N_n$ for Problem (P_a^N). In the general case, it is straightforward to show that the memory needs for $\mathbf{A}_{data}$ is just $(2N_x^2 + 2N_xN_u + N_u^2) \times N_n$, where N_x and N_u are the number of state and control variables respectively (in continuous time). That is, the growth is only linear with respect to N_n . Hence, the Hessian storage requirements belongs to the second column of Table 1, namely $\mathcal{O}(n)$. As an example, the computer memory for storing $\mathbf{A}_{data}$ for a million grid points for a typical aerospace guidance problem (i.e. $N_x = 6, N_u = 3$) is less than one GB of memory (~ 0.9 GB).

The computational cost of computing the product of $\mathbf{A}_{data}$ with a vector (see Eq. (2)) is $3N_n$ by inspection of Fig. 2. In the general case, the computational cost is approximately $3N_x^2 \times N_n$. Hence, the computational speed of computing a matrix-vector product with $\mathbf{A}_{data}$ as the matrix is just $\mathcal{O}(N_n)$. From the fourth column of Table 1, it follows that this computation can be done in tens of microseconds even when the grid is one million points.

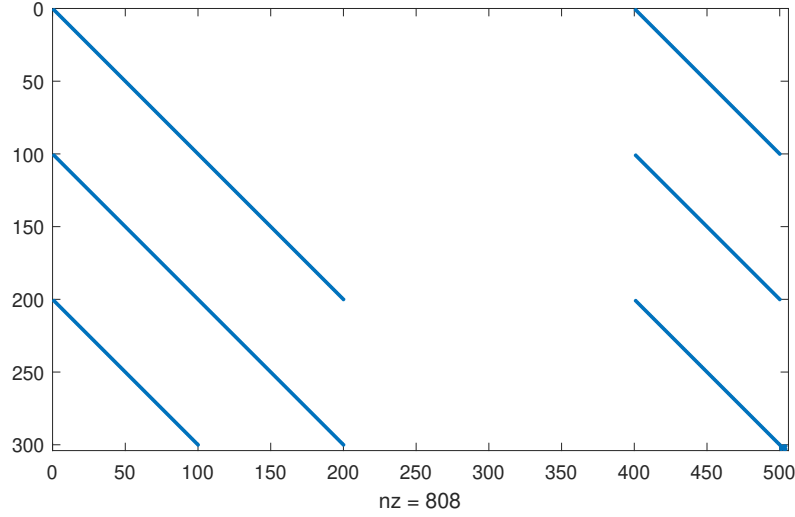


Figure 2. Data-dependent sparsity pattern of A_{data}

It is clear that neither space nor time complexity associated with A_{data} is a present-day obstacle to the challenge posed in [12] of producing a real-time, million-point trajectory optimization solver. More importantly, because the contents of A_{data} are problem specific, the preceding analysis is valid across the scope of all trajectory optimization problems. The origin of this powerful conclusion is due to the following two facts: (i) A universal Birkhoff-theoretic method decouples the discretization from the problem-specific data and (ii) the problem-specific data is sampled pointwise with no dependencies on the adjacent grid points. The last point is in sharp contrast to Runge-Kutta discretization methods that have interdependencies across the grid points [26].

B. Computational Cost Associated with the Data-Independent Matrix, A_0

The sparsity pattern of A_0 is shown in Fig. 3. The dense blocks in this figure correspond to the

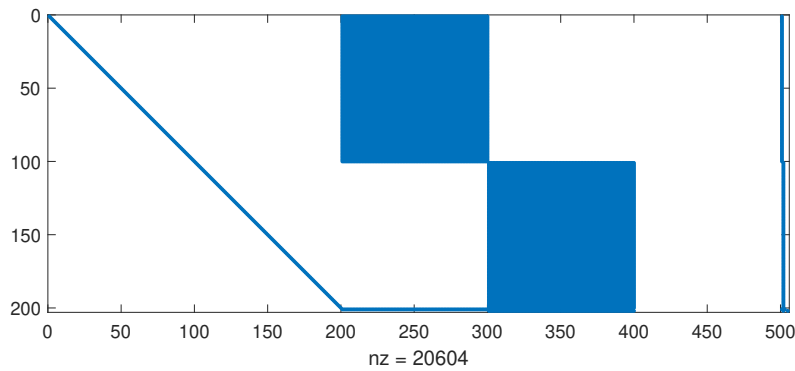


Figure 3. Data-independent sparsity pattern of A_0

B^a and B^b blocks of A_0 . At first glance, it appears that the Birkhoff matrices might be the main source of a computational burden. To better process this initial intuition, we break up our analysis into two extreme cases of Table 1, namely rows 1 and 4.

1. Thousands of Grid Points

The first row of Table 1 indicates that storing a Birkhoff matrix for a grid of one thousand points takes only fractions of a gigabyte (0.008 GB). Furthermore, solves of Eq. (2) are milliseconds per iteration even if they are performed at the slowest rate of $O(n^3)$. See top right corner of Table 1. From the second row of Table 1, it follows that computations over grids of less than 10,000 points can also be performed at moderately fast speeds using only a few gigabytes of memory. In other words, computations over a few thousand grid points are not a serious obstacle to a Birkhoff-theoretic method for trajectory optimization. Coupled with the fact that a universal Birkhoff method has $\mathcal{O}(1)$ condition numbers and a theoretical infinite rate of convergence [8], it follows from Table 1 that many aerospace trajectory optimization problems can be solved rapidly and accurately over thousands of grid points.

2. Addressing the Million Grid-Point Challenge

The challenge of solving a million point trajectory optimization problem was posed in [12] as an enabler for advancing a new theory for solving discrete-continuous mission planning problems [27–29]. From row 4 of Table 1, it follows that storing a Birkhoff matrix requires 8 terrabytes of memory. Furthermore, solves of Eq. (2) require a million seconds (≈ 11 days) per iteration at $\mathcal{O}(N_n^3)$ speed. Obviously, these numbers indicate a severe technological barrier. However, it was shown in [30] that if the Chebyshev version of the Birkhoff matrix is chosen (see Fig. 1) then solves of Eq. (2) can be performed at $\mathcal{O}(N_n \log N_n)$ computational speed (see column 5 of Table 1) using matrix-free techniques [31]. The space complexity of this new approach is $\mathcal{O}(N_n)$ (column 2 of Table 1). In other words, the matrix-free, Chebyshev version of the universal Birkhoff-theoretic method holds the potential to solve the million-point trajectory optimization challenge problem [12] in fractions of a second using only a few gigabytes of memory.

VII. A Brief Discussion on Alternative Birkhoff Hessians

The Hessian associated with a generic NLP is given by [1–3],

$$\partial^2 L(\mathbf{y}, \boldsymbol{\psi}) := \begin{bmatrix} L_{y,y} & L_{y,\psi} \\ L_{\psi,y} & 0 \end{bmatrix} \quad (30)$$

where $(\mathbf{y}, \boldsymbol{\psi})$ is the primal-dual pair. Choosing this pair by setting $\mathbf{y} = (\tilde{X}, \tilde{V}, \tilde{U}, x^a, x^b)$, $\boldsymbol{\psi} = (\Lambda, \Omega, \nu, \lambda^b)$ and reusing the special weighted Lagrangian given by Eq. (9) we get the following matrix (written in a Hamiltonian form):

$$\tilde{\mathbf{A}} := \left[\begin{array}{ccccc|cccc} W_B^{-1}H_{X,X} & 0 & W_B^{-1}H_{X,U} & 0 & 0 & H_{X,\Lambda} & I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -I & \mathbf{B}^b & 0 & \mathbf{b} \\ W_B^{-1}H_{U,X} & 0 & W_B^{-1}H_{U,U} & 0 & 0 & H_{U,\Lambda} & 0 & 0 & 0 \\ 0 & 0 & 0 & \bar{E}_{a,a} & \bar{E}_{a,b} & 0 & -\mathbf{w}_B^T & \bar{E}_{a,\nu} & 1 \\ 0 & 0 & 0 & \bar{E}_{b,a} & \bar{E}_{b,b} & 0 & 0 & \bar{E}_{b,\nu} & -1 \\ \hline H_{\Lambda,X} & -I & H_{\Lambda,U} & 0 & 0 & 0 & 0 & 0 & 0 \\ I & (\mathbf{B}^b)^T & 0 & -\mathbf{w}_B & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \bar{E}_{\nu,a} & \bar{E}_{\nu,b} & 0 & 0 & 0 & 0 \\ 0 & \mathbf{b}^T & 0 & 1 & -1 & 0 & 0 & 0 & 0 \end{array} \right] \quad (31)$$

Note the following:

1. Unlike $\mathbf{A}$ (Cf. Eq. (28)) which is a “ 10×10 ” matrix, $\tilde{\mathbf{A}}$ is “ 9×9 ”. This is because λ^a is absent in Eq. (31). As indicated in Remark 2, Eq. (13), and hence λ^a , does not exist in the context of an NLP. Instead, Eq. (12d) takes the following form,

$$\lambda^b - \mathbf{w}_B^T \Omega + \partial_{x^a} \bar{E}(\nu, x^a, x^b) = 0$$

2. Unlike $\mathbf{A}$ which is not symmetric, $\tilde{\mathbf{A}}$ is indeed symmetric.
3. The bottom left block of $\tilde{\mathbf{A}}$ is not the Jacobian of the constraints as is the case with an unweighted Lagrangian.
4. A direct application of Gershgorin’s theorem to $\tilde{\mathbf{A}}$ will not yield Theorem 1.
5. An application of Gershgorin’s theorem to $\tilde{\mathbf{A}}$ will yield eigenvalue estimates that are grid dependent. See, for example, the last column/row.

Now suppose a standard unweighted Lagrangian is used instead of the special one given by Eq. (9). Retracing all the steps taken in Sections III and IV, it can be observed (see also [8] for additional details) that Eq. (10) generates a weak implementation of the Hamiltonian conditions indicated in Eq. (22). To better appreciate why a weak implementation may not be desirable, consider its implications on Eq. (22d). Its weak implementation would take the form,

$$\mathbf{w}_B \circ \partial_u H(\Lambda, X, U) = 0 \quad (32)$$

Because it can only be imposed up to a practical tolerance of $\delta > 0$, the computational version of Eq. (32) takes the form,

$$\mathbf{w}_B \circ \partial_u H(\Lambda, X, U) \leq \delta[1, \dots, 1]^T \quad (33)$$

Hence the dual computational constraint that effectively gets imposed by Eq. (33) can be written as,

$$\partial_u H(\lambda_i, x_i, u_i) \sim \mathcal{O}(\delta/w_i), i = 0, \dots, N \quad (34)$$

Because $0 < w_i < 1, i = 0, \dots, N$, the effective tolerance on imposing Pontryagin’s Hamiltonian minimization condition is greater than δ , with the “worst” impositions occurring at the extreme points, i.e., 0 and N . This is because w_0 and w_N are the smallest of all the numbers $0 < w_i < 1, i = 0, \dots, N$ [7]. In other words, $u_i, i = 0, \dots, N$ resulting from solving Eq. (34) would be inaccurate (relative to an imposition of Eq. (22d)) with the worst accuracy near the boundary points. For these reasons, a standard Lagrangian is not computationally efficient, particularly for increasing values of N . The alternative is, of course, to use a weighted Lagrangian.

We briefly note that the argument for a weighted Lagrangian is not limited to Birkhoff-theoretic methods; it has been discussed before [16, 23] in the context of Lagrange pseudospectral methods (i.e., pseudospectral methods based on differentiation matrices [16]) and Runge-Kutta methods [23].

As a final point of discussion, consider once again retracing all the steps taken in Sections III and IV. It can be easily observed by this process that the source of asymmetry in $\mathbf{A}$ is the introduction of Eq. (13) and the use of the identity,

$$\mathbf{b} \equiv \mathbf{W}_B^{-1} \mathbf{w}_B \quad (35)$$

Both Eq. (13) and Eq. (35) are necessary to cast the Lagrange form of the necessary conditions (i.e., Eq. (10)) to an “equivalent” Pontryagin Hamiltonian form given by Eq. (22). In choosing to write

a Hessian in its strong Hamiltonian form with both initial and final transversality conditions, the symmetry of Eq. (31) is lost in Eq. (28). Nevertheless, note that the asymmetric and strong Hessian given by Eq. (28) is more meaningful in directly connecting it to Pontryagin's Principle [15] in at least two ways: (i) the explicit use of both the initial and final transversality conditions and (ii) a direct connection to the Pontryagin Hamiltonian without an intervening weight function; see also Eq. (29).

Because the entire body of literature in NLPs is associated with a symmetric Hessian, it is apparent that new algorithms beyond NLPs are needed to take full advantage of the properties associated with Eq. (28) such as those identified in Theorem 1. This is an open area of research.

VIII. Conclusions

In much of the literature on trajectory optimization, it is more common to see a detailed analysis of the Jacobian and its sparsity pattern rather than the Hessian. This is likely because in typical (non-Birkhoff) discretizations of optimal control problems, a detailed analysis of the Hessian is quite foreboding. For example, in Runge-Kutta methods, functions must be evaluated at collocation points that are not the discretization points. In sharp contrast, a universal Birkhoff-theoretic method decouples the discretization from the problem data functions through a natural introduction of virtual and co-virtual control variables. These primal and dual virtual variables appear linearly in the discretized optimal control problem and hence facilitate a substantially simpler process of analyzing a Hessian at a detailed granular level through the use of the covector mapping principle. The main “trick” employed to standardize the analysis in this note is to utilize the discretized Pontryagin Hamiltonian and its derivatives. Furthermore, the nonlinear components of the Hessian do not contain any Birkhoff-specific elements. This makes a Birkhoff Hessian more amenable to a detailed analysis. All of these desirable features point toward additional advantageous attributes of the universal Birkhoff theory that go well beyond its previously-discussed properties of flat condition numbers. Obviously, much remains to be done in advancing the theory for fast trajectory optimization. Nonetheless, the results obtained so far lend further credibility to the notion that the universal Birkhoff theory offers a disruptive method for rapidly solving nonlinear, nonconvex trajectory optimization problems.

Acknowledgments

The author gratefully acknowledges partial funding for this research provided by the Air Force Office of Scientific Research (AFOSR) and the Defense Advanced Research Project Agency (DARPA). The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the AFOSR, or DARPA or the U.S. Government.

References

- [1] Luenberger, D. G. and Ye, Y. *Linear and Nonlinear Programming*, Springer, 2008, Ch. 1, Ch. 15.
- [2] Nocedal, J. and Wright, S. *Numerical Optimization*, Springer, 2006. Chs. 2, 12, 15, 18.

- [3] Bazaraa, M. S., Sherali, H. D. and Shetty, C. M. *Nonlinear Programming: Theory and Algorithms*, Wiley-Interscience, 2006. Chs. 4, 7 10.
- [4] G. Golub and C. F. Van Loan, *Matrix Computations*, 4th Edition, The Johns Hopkins University Press, Baltimore, MD, 2013. Chs. 1, 3, 4, 11.
- [5] Nesterov, Y., *Introductory Lectures on Convex Optimization: A Basic Course*, Kluwer Academic Publishers, 2004. Ch. 1.
- [6] L. N. Trefethen and D. Bau, III, *Numerical Linear Algebra*, SIAM, Philadelphia, PA, 1997. Chs. 1, 3, 4, 6.
- [7] I. M. Ross, R. J. Proulx, C. F. Borges, “A Universal Birkhoff Pseudospectral Method for Solving Boundary Value Problems,” *Applied Mathematics and Computation*, Vol. 454, 128101, 2023. <https://doi.org/10.1016/j.amc.2023.128101>
- [8] Ross, I. M., “A Universal Birkhoff Theory for Fast Trajectory Optimization,” *Journal of Guidance, Control and Dynamics*, vol. 47, no. 12, pp. 2468–2481, Dec. 2024. <https://doi.org/10.2514/1.G007737>
- [9] Proulx R. J. and Ross, I. M., “Implementations of the Universal Birkhoff Theory for Fast Trajectory Optimization,” *Journal of Guidance, Control and Dynamics*, vol. 47, no. 12, pp. 2482–2496, Dec. 2024. <https://doi.org/10.2514/1.G007738>
- [10] Kreyszig, E. *Advanced Engineering Mathematics*, Wiley, 2011. Ch. 20.
- [11] Arora, S. and Barak, B. *Computational Complexity: A Modern Approach*, Cambridge University Press, 2007. Ch. 4.
- [12] Ross, I. M., “The Million Point Computational Optimal Control Challenge,” *SIAM Conference on Control and its Applications*, MS24, July 10–13, 2017, Pittsburgh, PA.
- [13] I. M. Ross, Q. Gong, M. Karpenko and R. J. Proulx, “Scaling and Balancing for High-Performance Computation of Optimal Controls,” *Journal of Guidance, Control and Dynamics*, Vol. 41, No. 10, 2018, pp. 2086–2097. <https://doi.org/10.2514/1.g003382>
- [14] I. M. Ross, “Enhancements to the DIDO Optimal Control Toolbox,” arXiv preprint, arXiv:2004.13112, 2020. <https://doi.org/10.48550/arXiv.2004.13112>
- [15] I. M. Ross, *A Primer on Pontryagin’s Principle in Optimal Control*, Second Edition, Collegiate Publishers, San Francisco, CA, 2015. <https://www.amazon.com/Primer-Pontryagins-Principle-Optimal-Control/dp/0984357114>.
- [16] F. Fahroo and I. M. Ross, “Advances in Pseudospectral Methods for Optimal Control,” *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2008-7309, Honolulu, Hawaii, August 2008. <https://doi.org/10.2514/6.2008-7309>
- [17] F. Fahroo and I. M. Ross, “Pseudospectral Methods for Infinite-Horizon Optimal Control Problems,” *Journal of Guidance, Control and Dynamics*, Vol. 31, No. 4, pp. 927–936, 2008. <https://doi.org/10.2514/1.33117>

- [18] N. Koeppen, I. M. Ross, L. C. Wilcox and R. J. Proulx, “Fast Mesh Refinement in Pseudospectral Optimal Control,” *Journal of Guidance, Control, and Dynamics*, vol. 42 no. 4, pp. 711–722, 2018. <http://dx.doi.org/10.2514/1.G003904>
- [19] I. M. Ross and R. J. Proulx, “Further Results on Fast Birkhoff Pseudospectral Optimal Control Programming,” *Journal of Guidance, Control, and Dynamics*, vol. 42 no. 9, pp. 2086–2092, 2019. <http://dx.doi.org/10.2514/1.G004297>
- [20] B. A. Conway, “A Survey of Methods Available for the Numerical Optimization of Continuous Dynamic Systems,” *Journal of Optimization Theory and Applications*, Vol. 152, 2012, pp. 271–306. <https://doi.org/10.1007/s10957-011-9918-z>.
- [21] J-B. Caillaud, R. Ferretti, E. Trélat, and H. Zidani, “An algorithmic guide for finite-dimensional optimal control problems,” *Chapter 15, Handbook of Numerical Analysis*, Elsevier, Volume 24, 2023, Pages 559–626, <https://doi.org/10.1016/bs.hna.2022.11.006>.
- [22] M. Karpenko, J. K. Halverson, Y. Tedla, T. Lippman, K. Han, I. M. Ross, M. K. Barker, M. S. Robinson, W. Carter, A. K. Boyd, G. W. Patterson, C. M. O’Shea, F. S. Turner, R. T. Poffenbarger, C. J. Dennehy, T. Vanzwieten, and A. A. Wolf, “Fast Attitude Maneuvers for NASA’s Lunar Reconnaissance Orbiter: Practical Flight Application of Attitude Guidance Using Birkhoff Pseudospectral Theory and Hamiltonian Programming,” *IEEE Control Systems Magazine*, Vol. 44, No. 2, 2024, pp. 26–54. doi: 10.1109/MCS.2024.3358593.
- [23] I. M. Ross, “A Direct Shooting Method is Equivalent to An Indirect Method,” arXiv preprint, arXiv:2003.02418, 2020. <https://doi.org/10.48550/arXiv.2003.02418>
- [24] Bryson, A. E., and Ho, Y.-C., *Applied Optimal Control*, Hemisphere, New York, 1975 (Revised Printing; original publication, 1969). Ch. 2.
- [25] Hennessy, J. L. and Patterson, D. A., *Computer Architecture: A Quantitative Approach*, Sixth Edition, Morgan Kaufmann Publishers, Cambridge, MA, 2019. Ch. 1
- [26] Ascher, U. M. and Petzold, L. R., *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, SIAM, Philadelphia, PA, 1998. doi = 10.1137/1.9781611971392.
- [27] I. M. Ross, M. Karpenko and R. J. Proulx, “A Nonsmooth Calculus for Solving Some Graph-Theoretic Control Problems,” *IFAC-PapersOnLine*, Volume 49, Issue 18, 2016, Pages 462–467, <https://doi.org/10.1016/j.ifacol.2016.10.208>.
- [28] I. M. Ross, R. J. Proulx and M. Karpenko, “Autonomous UAV Sensor Planning, Scheduling and Maneuvering: An Obstacle Engagement Technique,” *2019 American Control Conference (ACC)*, Philadelphia, PA, USA, 2019, pp. 65–70, doi: 10.23919/ACC.2019.8814474. <https://ieeexplore.ieee.org/document/8814474>.
- [29] I. M. Ross and R. J. Proulx and M. Karpenko, “An Optimal Control Theory for the Traveling Salesman Problem and Its Variants,” arXiv preprint, arXiv:2005.03186, 2020, <https://arxiv.org/abs/2005.03186>.

- [30] A. Javeed, D. Ridzal, D. Kouri and I. M. Ross, “A Fast Matrix-Free Method for Low-Thrust Trajectory Optimization,” *AAS/AIAA Astrodynamics Specialist Conference and Exhibit*, Big Sky, MT, August 13-17, 2023, Paper No. AAS-280.
- [31] Peng, R. and Vempala, S. S., “Solving Sparse Linear Systems Faster than Matrix Multiplication,” *Communications of the ACM*, 67/7, 2024, pp. 79–86. <https://doi.org/10.1145/3615679>.