AAS 05-332

A HISTORICAL INTRODUCTION TO THE COVECTOR MAPPING PRINCIPLE

I. Michael Ross *

Abstract

In 1696, Johann Bernoulli solved the brachistochrone problem by an ingenious method of combining Fermat's principle of minimum time, Snell's law of refraction and "finite element" discretization. This appears to be the first application of a "direct method." By taking the limits of these "broken-line solutions," Bernoulli arrived at an equation for the cycloid. About fifty years later (1744), Euler generalized Bernoulli's direct method for the general problem of finding optimal curves and derived the now-famous Euler-Lagrange equations. Lagrange's contribution did not come until 1755 when he (Lagrange) showed that Euler's result could be arrived at by an alternative route of a new calculus. Lagrange's ideas superceded the Bernoulli-Euler method and paved the way for a calculus of variations that culminated in the 1930s at the University of Chicago. In the late 1950s, the complexity of these variational equations were dramatically reduced by the landmark results of Bellman and Pontryagin. Their results are connected to Karush's generalization of Lagrange's yet-another-idea of "undetermined" multipliers. The simplicity of their equations also came with an amazing bonus of greater generality that engineers could now conceive of applying their results to practical problems. In recognizing that the elegant methods of Bellman and Pontryagin were not scalable to space trajectory optimization, astrodynamicists developed a broad set of computational tools that frequently required deep physical insights to solve real-world mission planning problems. In parallel, mathematicians discovered that the equations of Bellman and Pontryagin were incompatible with the original ideas of Bernoulli and Euler. Since the 1960s, intense research within the mathematical community has lead to the notion of "hidden convexity," set-valued analysis, geometric integrators and many other mathematical topics that have immediate practical consequences, particularly to simplifying complex mission planning problems. This is the story of the covector mapping principle. When combined with a modern computer, it renders difficult trajectory optimization problems remarkably easy that it is now possible to routinely generate even real-time solutions.

INTRODUCTION

As this is not a typical AAS research paper, I will take the somewhat unusual route of writing this article in first person. This is a history paper. It is a fascinating story about many, apparently disparate topics, that have come together only in recent years because of interdisciplinary activities. What is remarkable about this story is how difficult topics get simplified by way of new ideas† ... and that these new ideas were indeed a path once taken and abandoned by none other than Bernoulli and Euler before Lagrange "killed" it with his new calculus.

I will tell this story from a somewhat chronological perspective but interweave it with some modern ideas. Told from a strict chronological perspective without these insights, the story has no plot and the topics seem too abstract. With 20/20 hindsight, the story is almost predictable, and the plot may appear simple to the naïve. Because the right perspective is important to appreciate this narrative, I find Columbus' story[1] helpful to set the stage:

> During a dinner in his honor, the gentlemen in attendance belittle Columbus' feats by suggesting that anybody can sail across the ocean and find land — it is the simplest thing to do! After a while, Columbus asks the men if they could make an egg stand upright. After several failed attempts, they declare it is an impossible feat. Columbus

*Professor, Department of Mech. & Astro., Code ME/Ro, Naval Postgraduate School, Monterey, CA 93943; imross@nps.edu. Associate Fellow, AIAA.

†By Ockham's Razor, these new ideas must be the truth model.

then simply taps the egg to crack its base and makes it stand upright quite easily. When the men protest that they didn't know they could crack the egg, Columbus replies, "It's easy after I've shown you how."

In commemoration of the claim that Sant Antoni de Portmany in the island of Ibiza (Spain) is the birthplace of Columbus, a statue of "The Egg" shown in Fig. 1 was erected

Figure 1: The Egg of Columbus in Ibiza.

in the early 1990s. Its center contains a model of a 15^{th} century ship.

Solving trajectory optimization problems today is quite easy because we know how to do it. But the challenges we face today are quite different from those of yesteryears[2] precisely because of our own success in routinely solving a very large number of problems in trajectory optimization, and many in real-time as well. What has made this practically possible over the last few years is the proper blending of emerging sets of mathematical tools. Prior attempts on this effort, primarily in the 1960s, failed because even though researchers then had a "draft of the plot," the blending was done somewhat early from the points of view of both science and technology. What made this possible in the late 1990s, was the near-simultaneous confluence of three major breakthroughs:

1. The widespread availability of extraordinary computer technology on ordinary computers;
2. Global convergence of nonlinear programming (NLP) techniques (both, theory and software); and,
3. An extension of the Bernoulli-Euler ideas in the form of the covector mapping principle (CMP).

While even non-engineers are aware of the remarkable progress in computer technology, many astrodynamicists are not fully aware that current NLP algorithms are indeed globally convergent (see for example, Ref. [3]) ... or that NLP techniques are embedded in equation solvers, implicit methods, shooting methods and a host of other techniques[4] that constitute the standard arsenal of a practicing engineer. That many practitioners do not even recognize that they implicitly use NLP techniques is a testament to its widespread acceptance as stock technology. Thanks to problem solving environments like MATLAB$^{\text{TM}}$— sophisticated mathematics can now be performed with very few lines of code. Nonetheless, this is largely a story of the third item indicated above. When combined with items 1. and 2., the CMP renders hard problems easy in the sense that engineering problems that were once considered difficult can now be solved routinely. As a stand-alone topic, the mathematics of the CMP is set-valued analysis in Sobolev spaces[5]. Although this mathematics is new (at least with regards to engineering), it it actually very intuitive because it is also very practical. This means that the techniques of trajectory optimization are now more firmly rooted in first principles than ever before[6].

Rather than delve right into all these issues and their relationship to astrodynamics, consider for the purposes of argument an apparently trivial problem. Let r be a positive *irrational* number less than 1, $x(0) = 0 = t_0$, and $t_f = 1$. The trajectory optimization problem is to,

$$\left.\begin{array}{ll} \textsf{Minimize} & [x(t_f) - r]^2/2 \\ \textsf{Subject to} & \dot{x}(t) = u(t) \\ & u = 0 \text{ or } 1 \end{array}\right\} \tag{1}$$

This problem is a modification of one of the counter examples discussed in Ref. [7]. Even without the aid of Pontryagin's Minimum Principle (PMP), it is clear that any feasible control function, $u(\cdot)$, that takes the state of the system from $x = 0$ to $x = r$ in unit time is a globally minimizing solution. This *infinite family of globally optimal solutions* is indicated in Fig. 2. An application of the PMP

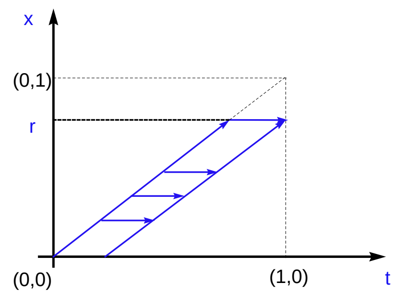

Figure 2: An illustration of the infinite family of globally optimal solutions to Problem 1.

generates the same result as our intuition suggests.

Now, let's pretend that we don't know the solution to this problem but know how to write the necessary conditions arising from the PMP. We then design an algorithm, say a shooting method, to solve for these necessary conditions. This is known as the "indirect" shooting method in the literature[8]. That is, we integrate the resulting state and costate equations over N, nonzero step sizes, h, so that $Nh = 1$, the final time. Since N is an integer, $h = 1/N$ is rational. This means that as the integration proceeds according to, $x_k = x_{k-1} + hu_k$, all the states at the time steps are rational. Consequently, x_N, which is expected to approximate $x(t_f)$, is rational for all step sizes, h, no matter how small. Thus, $x_N \neq r$; or equivalently,

$$x_N - r > 0 \quad or \quad x_N - r < 0$$

Combining this result with the transversality conditions in conjunction with the discretized adjoint equations, it is quite straightforward to show that,

$$\lambda_k = x_N - r, \ \forall \ k$$

where λ is the costate. This means that $\lambda_k \neq 0, \forall \ k$. By applying the rest of the PMP we come to the *erroneous conclusion that there is no extremal solution to Problem 1* for any step size, h, and hence an optimal solution does not exist! Note that there is no digital computer intervening here. In fact, we did this exercise completely by hand without introducing any "round off errors;" that is, the computations are exact.

What is even more amazing about Problem 1 is that if we were to solve it "directly;" that is, without the use of Pontryagin's principle, we would indeed get a solution! This solution approximates the exact solution as shown in Fig. 3. *Thus, contrary to popular belief, indirect methods are not more accurate than direct methods.* In fact, Problem 1 has multiple globally optimal solution and a direct method generates all of them (within precision h) while an indirect method produces not even one.

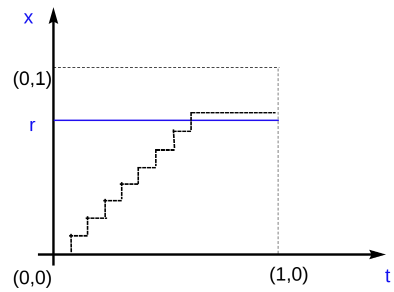

Figure 3: An approximate solution to Problem 1 by a direct (Bernoulli-Euler) method. The multiplier or indirect method generates no solution.

Because Problem 1 is simple and easy to understand, all the points I've noted above seem strange because they are not widely known within the engineering community despite that they are folklore in the mathematical community[9, 10]. While some engineers may dismiss Problem 1 as a clever mathematical trick, I will now show why this problem contains the many ingredients of the common misconceptions in engineering trajectory optimization. Here is a small sample of some popular *misconceptions*:

1. Indirect methods are more accurate than direct methods (already disproved by Problem 1);
2. Indirect methods do not require nonlinear programming techniques;
3. A good guess is required to solve trajectory optimization problems;
4. Low-thrust trajectory optimization problems are hard problems; and,
5. Soft computing methods (such as genetic algorithms and simulated annealing) produce globally optimal solutions.

There are many other popular misconceptions, and some of them do have justifiable origins. That is, some of the misconceptions are based on certain assumptions, but these assumptions are no longer true due to recent developments in mathematics and software. The "trick" is that much of these advances require a new line of thinking. For example, the well-known sensitivity problem[11] associated with a Hamiltonian system is due to integrating the differential system by "propagating" the initial conditions. If there is no propagation, there is no sensitivity problem. That it is indeed possible to solve differential equations without propagating them implies that the symplectic structure of the Hamiltonian system (which is the root cause of the sensitivity problem) is no longer an obstacle to problem solving. The theory behind this radical viewpoint is based on treating a differential system as a generalized (set-valued) equation[5]. In practical terms, this means that such a system must be solved by "batch" methods and not by recursion, in sharp contrast to conventional wisdom. Batch methods for differential systems were impractical prior to the advent of large-scale numerical methods; hence, they were largely pursued by mathematicians for theoretical functional analysis; see Refs.[9] and [10] and the references contained therein. At the turn of the 20^{th} century, it became possible to solve problems with thousands of variables and constraints in real time, while million variable problems can now be solved routinely[‡] (see for example, Ref.[12]). Thus, abstract set-valued analysis can be intimately linked to practical problem solving. This brings into focus a new paradigm for both theory and practice. The confluence of all these factors in the late 1990s paved the way for a dramatic turn in simplifying practical problem solving. That these simplifications can be traced back to Bernoulli and Euler is truly a testament to their genius.

[‡]Consider for example, that 1 million variables require less than 8MB of storage.

THE ORIGINS

The brachistochrone problem is one of the most popular problems discussed in textbooks because it has a universally surprising conclusion: the minimum-time path for a falling bead is not a straight line. Having sided with Leibniz in the Newton-Leibnitz feud over the invention of calculus, Bernoulli's less-than-honorable intentions were aimed at exposing Newton and his method of fluxions (calculus) while simultaneously exalting himself to greatness[13]. In 1696, in the June issue of the journal, *Acta Erudotorium*, Bernoulli wrote[14],

> *I, Johann Bernoulli, address the most brilliant mathematicians in the world. Nothing is more attractive to intelligent people than an honest, challenging problem, whose possible solution will bestow fame and remain as a lasting monument. Following the example set by Pascal, Fermat, etc., I hope to gain the gratitude of the whole scientific community by placing before the finest mathematicians of our time a problem which will test their methods and the strength of their intellect. If someone communicates to me the solution of the proposed problem, I shall publicly declare him worthy of praise.*

I don't know of any present-day journal that would dare publish anything like this. In any case, what ensued after Bernoulli's challenge is beautifully retold by Sussmann and Willems[15]. Thus began the ignominious birth of optimal control theory. It spawned many other branches of mathematics with nonsmooth analysis[16] being its newest entry. Of great interest to me, and perhaps to all engineers, is not so much the solution to the brachistochrone problem (i.e. the cycloid) but how Bernoulli arrived at it; see Ref. [13] for details.

First, Bernoulli discretized the space into "finite elements" as shown in Fig. 4. Thus, the idea

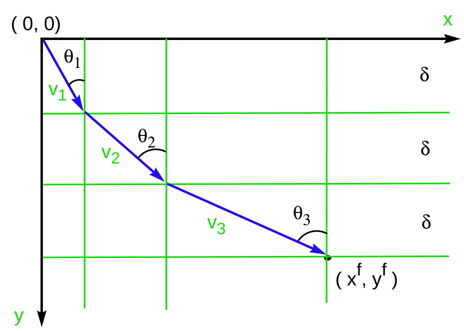

Figure 4: Bernoulli's discretization method for solving the brachistochrone problem.

of discretizing space and finding "discrete" solutions is not at all new; it's just that we do this more often with a digital computer. Also, Bernoulli did not invent this concept; it goes back to ancient mathematicians ... for example, estimating π by approximating a circle to a polyhedral[17]. *Also, recall that the whole notion of epsilons and deltas in first-principles mathematics is founded upon the ideas of approximation.* That we do this everyday on a digital computer and yet don't really think in terms of first-principles is a testament to the transparency in analysis a computer brings; yet, a failure in understanding precisely these first principles has lead to some widely-held misconceptions that I've indicated before. In any event, Bernoulli approached the brachistochrone problem by observing that the speed of a falling bead can be written as a function of position,

$$v(x, y) = \sqrt{2gy/m}$$

Now imagine this speed to be the speed of "light." Then according to Fermat's law, light travels in minimum time. The minimum-time path is given by Snell's law of refraction (see Fig. 4),

$$\frac{\sin\theta_1}{v_1} = \frac{\sin\theta_2}{v_2} = \text{constant}$$

Hence, over each finite element, the minimum-time path of the bead must satisfy,

$$\frac{\sin\theta_k}{\sqrt{y_k}} = \text{constant} \quad \forall\ k = 1, 2 \ldots N \tag{2}$$

where $y_k = k\delta$ and $\sum_k y_k = y^f$, a given vertical distance. After additional manipulation of variables, and passing to the limit $\delta \to 0, N \to \infty$, Bernoulli obtained an equation for the cycloid. In parametric form, this is given by,

$$x(\phi) = a(\phi - \sin\phi) \qquad y(\phi) = a(1 - \cos\phi) \tag{3}$$

where a is a constant of integration.

In computerizing, Bernoulli's method, we would simply solve the problem for some N "sufficiently large." Thus, the drudgery of the many steps required to go from Eq.(2) to Eq.(3) is computerized.

It is a big mistake to assume that solving the problem for N sufficiently large is less accurate than Eq.(3). Just because Eq.(3) is written in terms of familiar functions does not mean that it can be computed exactly. Because this is such a widespread misconception, I will now devote a special section to this concept before returning to our main story.

EXACT SOLUTIONS ARE INACCURATE!

Suppose that three solutions to some problem are written as, $x_1 = \pi$, $x_2 = 3.14159$, and $x_3 \approx \pi$. Suppose that we regard x_1 as the exact solution. It remains a symbolic solution until it is computed. Considering that research on the computation of π has spanned 2000 years and continues to this day[17], I regard $x_1 = \pi$ to be an approximate solution masquerading as an exact solution. Thus, there is no difference between the "exact" solution, x_1, and the "approximate" solution, x_3, if the precise nature of the approximation is clarified. In fact, x_3 is a more honest representation of the solution. In this spirit x_2 is the most useful solution and makes the exact and approximate solutions equivalent.

Now consider the field of real numbers, $\mathbb{R}$. Any real number is either rational or irrational. According to Cantor, the set of all rational numbers, $\mathbb{Q}$, is denumerable[18]. This means that "almost all numbers are irrational." According to Lebesgue, the measure of any denumerable set is zero. This means that the entire contribution to a measure comes from irrational numbers. Thus, we have,

Theorem 1 (Cantor-Lebesgue) *Almost all solutions are approximate; all solutions requiring a digital computer are approximate.*

Now consider computing a solution for the brachistochrone problem from the equation for the cycloid. The data for the problem are the initial conditions, $(0, 0)$, and some given numbers, (x^f, y^f), the coordinates of the target point (see Fig. 4). Using this data, we have to compute a (see Eq.(3)) and then connect ϕ to time; after all, this was a minimum-time problem. Hence, Eq.(3) should be more accurately described as an intermediate step in solving the brachistochrone problem. In other words, from a practical point of view, we have (or Bernoulli did!) reduced the solution to the brachistochrone problem to solving a set of nonlinear algebraic equations for the given problem data; this is a root-finding problem. We solve such problems today by NLP techniques as it is essentially a problem of finding a feasible solution to a set of algebraic equations.

Now, even if we got lucky and ended up with nice rational numbers to some of the problem data and constants, consider the computation of the trigonometric functions in Eq.(3). A computation of these functions by hand or by a computer is done by approximations that are equivalent to[19],

$$x(\phi) = \sum_{i=0}^{n} \alpha_i(\phi) \qquad y(\phi) = \sum_{i=0}^{n} \beta_i(\phi) \tag{4}$$

where $\alpha_i(\phi), \beta_i(\phi)$ are polynomials where n is "sufficiently large." This n can be related to the N in Eq.(2), and hence both the "exact" and "approximate" solutions are equivalent even in the absence of a digital computer. In other words, even the computation of a purportedly exact solution is approximate. I have described more details on this philosophy in Ref.[6]. Suffices to say, we can almost always never compute an exact solution ... and more importantly, an exact solution (whatever it means) is completely unnecessary in astrodynamics given that so much of our knowledge (vehicle parameters, gravity model etc.) is imprecise. It is not that the quest for solutions in terms of well-known functions is misguided; my point is that too many researchers (particularly in the academic community) focus their energies in seeking "exact" solutions to simple problems rather than devise techniques for approximate solutions to complex problems. As a result of this lopsided research, practical trajectory optimization problems appear to be hard to the uninitiated.

A final point worth noting is the connection between approximations and the notion of feedback in control theory. Almost the entire theory of feedback control is based on the presumption that exact models for systems cannot be obtained. If exact solutions to exact problems were possible, feedback would be unnecessary. That approximations are inherent and fundamental essentially underpins the entire field of control theory.

EULER'S GENERALIZATION OF BERNOULLI'S METHOD

It is clear that Bernoulli's brilliant, albeit *ad hoc* method can today be described as a direct method[8], except that Bernoulli's direct method requires an extraordinarily good guess! Nonetheless, it is the first instance of an application of a direct method to solve an optimal control problem.

In an effort to generalize Bernoulli's ideas, Euler (Bernoulli's student) took the step of devising the Euler integration method (see the discussion following Problem 1 in the Introduction section) to solve the "most general problem" (general during Euler's days),

$$\left.\begin{array}{ll} \textsf{Minimize} & \displaystyle\int_{t_0}^{t_f} F(x(t), \dot{x}(t), t)\, dt \\ \textsf{Subject to} & x_0 = x^0 \\ & x_f = x^f \end{array}\right\} \tag{5}$$

By using Bernoulli's direct method with his (Euler's) integration scheme, Euler discretized Problem 5 to a problem of ordinary calculus, used Fermat's rule to get the minimum (i.e. setting derivatives to zero) and took limits (as $h \to 0$). The details of this process are not trivial and require some clever mathematics[9]. After this lengthy process, he arrived at the (limiting) equation,

$$\frac{d}{dt}\frac{\partial F}{\partial \dot{x}} - \frac{\partial F}{\partial x} = 0 \tag{6}$$

Although Euler's mathematical route to Eq.(6) is tortuous, the net result is simple: an equation that anyone, who is not as ingenious as Bernoulli, can apply to "any" problem. By generalizing and automating Bernoulli's procedure, Euler gave us Eq.(6) so that no one else need endure the arduous path to solving Problem 5. The hope is that the new problem resulting from an application of Eq.(6) is less painful to solve. When this process of discretization and taking limits is encoded in a computer software, it is clear that theory and practice can be intimately connected. This is the main reason why the modern practice of optimal control is more firmly rooted in first principles than ever before[6]. The absence of a computer makes Euler's method (i.e. discretization and taking limits) drudgery. Consequently, it is not surprising that upon receiving a letter from a 19 year old Lagrange on the derivation of Eq.(6) using the new concept of variations, "Euler abandoned his ideas, espoused that of Lagrange and renamed the subject the 'calculus of variations.' " The rest is history[20]. The pinnacle of this thread of history was in the 1930s where Bolza (a former student of Weierstrass), Bliss and others at the University of Chicago were formulating a complete set of necessary conditions for problems in the calculus of variations[21].

LAGRANGE, KARUSH AND MATHEMATICAL PROGRAMMING

In the early 1900s, the mathematics department at the University of Chicago was a dominant force in American mathematics with faculty consisting of Birkhoff, Bliss, Bolza, Halmos, Hildebrandt, Weil and others[21]. Since the difficult problems in those days lay in variational calculus, it is perhaps forgivable that the 1939 Master's thesis work of William Karush that generalized Lagrange's method of "undetermined" multipliers to inequality constraints went completely unnoticed until Kuhn rediscovered it in 1974 [22]. In modern terminology, these multipliers are part of the notion of covectors. Prior to 1974, there were only five citations to Karush's work. Kuhn notes that, "Karush, as a graduate student [at the University of Chicago] on the road to a Ph.D.and a career in research, never thought of publishing his masters thesis, and Graves [Karush's advisor] did not encourage him to do so." Thus began the birth of nonlinear programming as a mathematical field separate from the calculus of variations. Any NLP (or for that matter any finite-dimensional optimization problem) has three, *and only three ingredients*:

- A finite collection of decision variables (i.e. optimization variables), or a vector, $\mathbf{x}$
- A set of allowable values, $\mathbb{X}$, that $\mathbf{x}$ can take (i.e. constraint set); and,
- A means of ranking the decisions (i.e. cost function) given by a map, $\mathbf{x} \mapsto Y$.

Thus, any optimization problem can be formulated as,

$$(A) \quad \begin{cases} \text{Minimize} & Y(\mathbf{x}) \\ \text{Subject to} & \mathbf{x} \in \mathbb{X} \end{cases}$$

In nonlinear programming, $\mathbf{x} \in \mathbb{R}^{N_x}$, $\mathbb{X} \subset \mathbb{R}^{N_x}$ and $Y : \mathbb{R}^{N_x} \to \mathbb{R}$. In multi-objective problems, Y is a vector-valued function. In integer programming, $\mathbf{x} \in \mathbb{Z}^{N_x}$ and so on. *Problem A is central to all trajectory optimization problems regardless of the method employed.*

With Euler implicitly declaring the death of direct methods in the 18th century, the calculus of variations, having reached its peak in the 1930s was close to being a complete theory until its rebirth in the late 1950s with the pioneering works of Pontryagin and Bellman. Their research was motivated by practical problems and not pure mathematics. Dreyfus[23] and Gamkrelidze[24] trace the motivations and the history behind these two men and their techniques. Although their theories are now well regarded, it is a little unfortunate that the importance of both ideas were initially dismissed as nothing but repackaging of the classical calculus of variations[25]. This perception appears to linger on among those who witnessed these landmark achievements[26]. It appears that the Egg of Columbus applies to Bellman and Pontryagin as well. Recent updates to the Bellman and Pontryagin frameworks are described in Refs. [27] and [28] respectively. In any event, both theories connected for the first time, the role of mathematical programming in optimal control in the form of a Hamiltonian minimization condition (HMC). That is, to find the optimal control $\mathbf{u}(t)$ at any time, t, both Bellman and Pontryagin require a solution to the (finite dimensional) optimization problem in the parameter $\mathbf{u}$,

$$(\mathsf{HMC}) \quad \begin{cases} \underset{\mathbf{u}}{\text{Minimize}} & H(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{u}) \\ \text{Subject to} & \mathbf{u} \in \mathbb{U} \end{cases}$$

where $\mathbb{U}$ is the "control space," (typically, some subset of $\mathbb{R}^{N_u}$) and, $\boldsymbol{\lambda}$ is a covector (that can be related to the costate or the Bellman value function), H is the control Hamiltonian[§], considered a function of $\mathbf{u}$ alone in Problem HMC. That is, in order to develop the optimality conditions for an optimal control problem, it is necessary to solve an optimization problem (compare Problem HMC to Problem A) in the form of minimizing a control Hamiltonian as a cost function subject to constraints, $\mathbf{u} \in \mathbb{U}$. That is, Karush's work was now central to solving an optimal control problem.

[§]The control Hamiltonian (Pontryagin's Hamiltonian) is not the same as the classical Hamiltonian (Hamilton's Hamiltonian); the latter is obtained by a Legendre transform of the former.

Since NLPs do not have closed-form solutions, many optimal control problems in the 1960s were cleverly formulated in a manner that Problem HMC could be solved in closed form; for example, linear-quadratic optimal control problems were the rage in the 1960s. Since such simplified problems were largely inapplicable to space trajectory optimization problems, astrodynamicists sought to solve the full nonlinear optimal control problems[29, 30]. This lead to astrodynamicsts separating themselves from the "control engineers" with the former interested in any set of solutions (and hence, largely open-loop solutions) to nonlinear problems while the latter (lead by Kalman) were mostly interested in closed-loop solutions to linear problems.

DIVORCE IN THE 1960s

It is ironic that although the works of Bellman and Pontryagin unified calculus of variations under the new theory of optimal control, it also divided theorists and practitioners along separate lines. As already noted, one of these divisions came in the form of those seeking feedback solutions to simple problems (i.e. control theorists) while others were interested in any solution to difficult problems (i.e. trajectory optimizers). In parallel, mathematicians were grappling with the new problems in optimal control theory itself. One of this was the the invalidity of the PMP for discrete-time systems; see for example, Refs.[31, 32]. That is, mathematicians, noticed that the PMP did not hold for discrete-time systems without an added assumption of convexity on the state velocity set (hodograph). This meant that a computer solution obtained by applying the PMP was suspect if the state velocity set was not convex. This is, in fact the reason why the indirect method for Problem 1 produces erroneous results because the control space, $\mathbb{U}$, (i.e. the set of allowable controls) is non-convex. If this space is convexified by allowing u to take all values between 0 and 1, then one can get the "correct" answer to Problem 1; however, one can easily argue that the convexified problem is no longer Problem 1. *Because certain algorithms (e.g. genetic algorithms) work in true discrete space, they can easily generate incorrect answers to continuous problems.* While soft computing has its advantages, its incorrect applications to continuous problems have lead many to false claims. Nonetheless, barring its many other problems, certain indirect methods appeared to work (on computers). While mathematicians scorned such engineering papers as largely incorrect (see Ref.[31] and [33] and the references contained therein for an extensive discussion on this topic), carefully designed guidance theories based on discretized solutions were successfully applied, particularly to space programs of the 1960s.

This apparently subtle mathematical point has immediate practical consequences. If the PMP does not hold for discrete-time systems, how can computer solutions be trusted as they are fundamentally discrete? This issue is best visualized by the diagram shown in Fig. 5. Here, Problem P

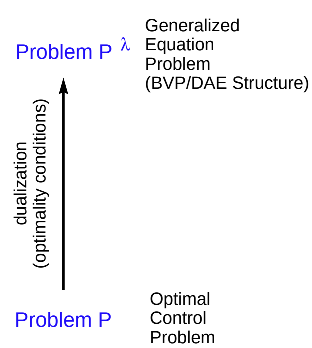

Figure 5: Application of the PMP to trajectory optimization problem.

represents a given trajectory optimization problem, say one that looks like:

$$
(\mathsf{P}) \begin{cases} \text{Minimize} & J[\mathbf{x}(\cdot), \mathbf{u}(\cdot), t_0, t_f] = \\ & E(\mathbf{x}_0, \mathbf{x}_f, t_0, t_f) + \int_{t_0}^{t_f} F(\mathbf{x}(t), \mathbf{u}(t), t) \, dt \\ \text{Subject to} & \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \\ & \mathbf{u}(t) \in \mathbb{U} \\ & (\mathbf{x}_0, \mathbf{x}_f, t_0, t_f) \in \mathbb{E} \end{cases}
$$

where $\mathbb{E}$ is some given endpoint set, $E : \mathbb{E} \to \mathbb{R},\ F : \mathbb{X} \times \mathbb{U} \times \mathbb{R} \to \mathbb{R},$ and $\mathbf{f} : \mathbb{X} \times \mathbb{U} \times \mathbb{R} \to \mathbb{R}^{N_x}$ are differentiable functions. The ideas to follow apply to nondifferentiable functions as well, but I will limit the discussion to nice smooth functions to avoid introducing nonsmooth analysis[16]. Now, by applying the PMP, one gets a "two-point" boundary value problem. Call this Problem P^λ. Problem P^λ is twice the dimension of the Problem P as a result of the costates whose dimension is exactly equal to that of the states. Hence, Problem P^λ can be considered as a "Pontryagin lift" of Problem P from dimension n to $2n$. Since Problem P^λ is typically unsolvable in closed-form, we seek to find "approximate" solutions to it by "approximation methods" that typically involves a computer. Because so many engineers forget that Runge-Kutta methods are approximations, they tend to use words like "exact" solution because it is good enough for engineering applications. Later, I will bring up the point that *if typical RK methods are used for trajectory optimization, it can lead to disastrous results*. This point is largely unknown in engineering optimization because it was discovered only in 2000 by Hager[34]. In any event, Problem $P^{\lambda N}$ in Fig. 6 represents the approximate solution to

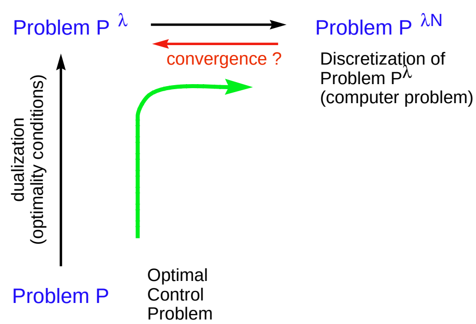

Figure 6: Schematic of an "indirect method" for trajectory optimization: incorrect results are possible when the state velocity set is nonconvex (i.e. most problems).

Problem P^λ where N denotes the number of points used, say for example, the number of points used in an RK method. So, whether you are an engineer or mathematician, you would want to know if as $N \to \infty$, the solution obtained from solving Problem $P^{\lambda N}$ approaches the solution to Problem P^λ. That is, you want the trajectory to be flyable. This is the problem of *convergence of the approximation*, and is not to be confused with convergence of the algorithm. Note also that we require convergence of the *solution*, not convergence of the *problem*. That is, just because Problem $P^{\lambda N}$ looks like a discrete approximation to Problem P^λ and both problems look the same as $N \to \infty$, *it does not mean* that the solutions converge. This is where the many engineering papers are flat wrong as Halkin[31] observed nearly forty years ago. Because many engineers still do not appreciate this point, there is widespread misconception on why trajectory optimization problems are hard to solve. I'll come back to this point later to explain why "convergent" RK methods do not converge for trajectory optimization problems. Note that these apparently subtle issues of correctly solving trajectory optimization problems are in addition to the well-known sensitivity problem of solving the Hamiltonian boundary value problem. These issues are not discussed in popular engineering texts on optimal control largely because some of the issues I've discussed above are relatively recent developments (late 1990s) although they were initiated during the 1960s. This is what I meant in the Introduction section in viewing history with a little bit of hindsight. So, we have two issues to contend with: even if we overcome the sensitivity problem by coming up with good guesses or by trial and error, the result cannot be trusted if the state velocity set is nonconvex¶.

While mathematicians were concerned about these issues, astrodynamicists moved on to apply the PMP to solve space trajectory optimization problems by using engineering judgement to answer the validity of the results. Because answers were not easy to come by (in part, because the theoretical issues were unresolved), trajectory optimization problems were declared hard problems. Thus, the issue of the validity of the PMP became mathematicians' problem. This is the reason why the topic of the discrete PMP is either omitted in many engineering texts, or even worse, erroneously stated.

IF BERNOULLI-HAD-A-COMPUTER METHODS

There is no doubt that the 1987 paper by Hargraves and Paris[35] remains one of the most influential papers on trajectory optimization. A quick citation search on various databases indicates well over 100 citations (as of 1 August 2005). Hargraves and Paris were influenced by the work of Dickmanns and Well[36] on the Hermite-Simpson discretization of Pontryagin's necessary conditions. This is Problem $P^{\lambda N}$ in my notation (see Fig. 6). They (Hargraves and Paris) then decided to apply the Hermite-Simpson discretization "directly" similar to the way Bernoulli and Euler solved problems (see Fig. 7). Of course, the key difference was the availability of a computer. This meant that unlike

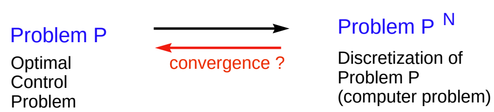

Figure 7: Schematic of a neo-classical Bernoulli-Euler, or "direct method:" incorrect results are possible even with "convergent" RK methods.

Bernoulli and Euler, taking limits was unnecessary ... a sufficiently large N would do the trick. While this may seem "obvious" to some, recall the story of the Egg of Columbus (see Fig. 1): it became obvious after Hargraves and Paris showed how to do it! Unlike prior works on collocation techniques[37], the Hargraves-Paris paper energized the trajectory optimization community because complex problems could now be solved with a grid size N much smaller than an Eulerian method; of course, $N = \infty$ is certainly not an option and is absolutely unnecessary for almost all purposes[6]; see also Theorem 1. That is, for theoretical purposes we need $N \to \infty$; for practical purposes, we only need to fully understand what happens as $N \to \infty$, but can get "exact" answers for N sufficiently large. Because of the emphasis on Eulerian methods in the 1960s and the limited computer capability, N could not be very large and hence the topic remained in the realm of theoretical mathematics. Although the concept of using higher-order methods, mesh refinement strategies etc. were known in the 1960s, the hard problems lay in solving large-scale NLPs if collocation methods were to be used. Recall that mathematicians were more concerned about the validity of the PMP for discrete-time systems rather than generating computer codes to solve problems. Besides, terraflop computation and petabyte storage was inconceivable. See Ref.[37] for a review of the early work. The Hargraves-Paris approach came at the right time: computing technology had now progressed to desktops and the Hermite-Simpson method offered fair accuracy for low N. Industrial strength NLPs were now available. Because of the limitations in computing technology of the late 1980s, N could not be very large (compared to 2005) and hence it lead to the misconception (among engineers) that direct collocation methods were inaccurate. It is a little unfortunate that this misconception persists to this day.

From the story so far, it appears that many engineers would have sought to obtain higher-order approximations earlier on so that they could solve practical problems with limited computational technology. Recall again that we are viewing history with a lot of hindsight. Regardless, one reason

¶Recall that any nonlinear equation generates a nonconvex set.

why this was not pursued in the 1960s and 70s was that NLP methods in those days were still struggling with convergence issues; *this is the convergence of the algorithm and is not to be confused with the convergence of the approximation*. The NLP bottleneck implied that there was no practical reason to pursue higher order methods. This is perhaps one reason why a 1976 paper by Hager[38] (a mathematician) on RK methods for optimal control received scant attention in both the mathematics and the engineering community. One reason the Hargraves-Paris paper was so influential was because the Hermite-Simpson method is relatively easy to use (compared to a standard RK4 method) and forms the basis of the software package, OTIS[39]. Despite advances in the higher-order RK type methods, the Hermite-Simpson method continues to be popular and remains a standard technique in SOCS[40] and other packages. It turns out that the Hermite-Simpson method is in fact a Runge-Kutta method[8, 40].

With the exception of Hager, mathematicians were largely uninterested in higher-order methods because it was incorrectly believed that all the mathematics of Eulerian methods would carry over to higher-order methods with just a higher rate of convergence. With the practical success of the Hermite-Simpson method, Conway and his students[41, 42] pursued an investigation of higher-order methods. In 1992, Enright and Conway[42] made a startling observation: by examining the Karush-Kuhn-Tucker (KKT) optimality conditions for the now-very-popular Hermite-Simpson method, they noticed that the KKT multipliers (i.e. the generalized Lagrange multipliers) did not seem to generate consistent discrete adjoint equations (see Fig. 8). Thus, the KKT multipliers would most certainly

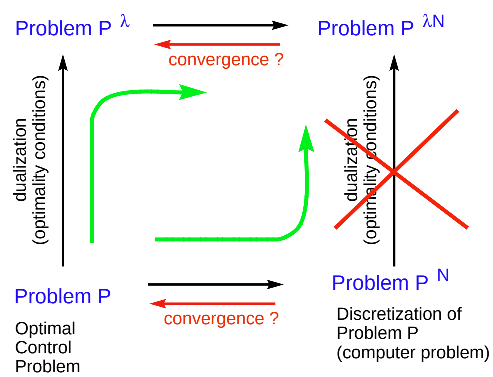

Figure 8: Schematic of a typical noncommutativity of direct and indirect methods, convergent or otherwise. Compare this with Figs. 7 and 6.

not converge to the costates, at least at not the same rate as the states — this was in sharp distinction to the Eulerian theory. While providing an excellent account of the issues involved, Enright and Conway also note that Hager[38] had outlined a theory for this behavior in 1976. Consequently, they recommended that the multiplier associated with the terminal transversality condition be back-propagated with a higher-order method (like RK4). This meant that if costates were desirable, one had to derive the adjoint equations and backward propagate it in a stable manner. von Stryk[43] recommended a simpler way of estimating the costates by comparing the limiting conditions of the KKT system to the continuous-time Pontryagin conditions. There is a very large literature in Europe on discrete methods. Due to lack of space, I will not discuss it here. In any event, an apparent consensus that emerged from such research was that a direct method be used for the initial solution of a trajectory optimization problem and this solution be refined by an indirect method using the estimates of the costates from the direct method. In other words, a "commutation gap" indicated in Fig. 8 was accepted as part of the sad facts of life.

TIPPING POINTS IN THE LATE 1990s

The late 1990s (the year 2000 for all practical purposes) were marked by the three tipping points noted in the Introduction section of this paper. While almost nobody remembers the "Y2K" non-problem, many trajectory optimization problems were being solved with much larger N (notably with Euler and Hermite-Simpson methods) as a result of the exponential growth in computer hardware and software. The optimization community regarded an optimal control problem as just a large-scale NLP and paid no attention to convergence of the approximation — all of effort was in the convergence of the algorithm. Furthermore, engineers considered an indirect method to be the "truth," so a two-stage process of using direct and indirect methods were being promulgated. In referring to Fig. 8, this meant that the solution obtained by solving Problem P^N be used as a guess to solving Problem $P^{\lambda N}$ to arrive at the "correct" answer. This was the state of the art in the mid 1990s.

In having made fundamental contributions to both NLPs and optimal control, Hager revisited his 1976 paper in an effort to exploit his adjoint transformation to properly integrate the computation of optimal controls with NLP techniques. From this emerged his landmark paper[34] in 2000. He combined the theory and practice of optimal control and explained, for the first time, many of the discrepancies in collocation methods. *Hager showed that if one were to use a typical RK method (based on the Butcher conditions), it would gloriously fail if it did not meet his new conditions that were in addition to those of Butcher.* This meant that very simple things like how the control was interpolated made a big difference between success and disaster. Conversely, it meant that simple fixes to existing methods were possible (i.e. theory still mattered!). Hager also showed that just because the KKT conditions for Problem P^N does not resemble the discretization of the Pontryagin conditions (i.e. Problem $P^{\lambda N}$), it does not mean that a direct method was inaccurate or that accurate costates could not be obtained. The missing ingredient was that a change in coordinates (i.e. a transformation) on the adjoints produced identical results as a indirect method, but these change in coordinates had to satisfy some additional conditions. This concept can be visualized as shown in Fig. 9. That is, even though the optimality conditions resulting from Problem P^N, denoted

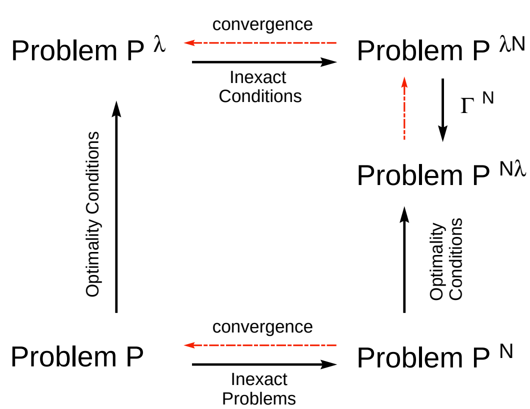

Figure 9: Schematic of the covector mapping principle.

as Problem $P^{N\lambda}$ are not necessarily the same as Problem $P^{\lambda N}$ (compare Fig. 8), these two problems can be made equivalent by a correct choice of discretization and a proper coordinate transformation. That is, the differential equation in an optimal control problem cannot be viewed as had been done in the past (i.e. primal space considerations alone), but its adjoint must be considered side-by-side regardless of the numerical method employed. *Thus, the Pontryagin conditions could not be ignored even in a direct method!* So, to design a correct method, we have to show the existence of an order-preserving[44] map, Γ^N, between the covectors (i.e. the complete set of multipliers, and not just the costates) of Problems $P^{\lambda N}$ and $P^{N\lambda}$ (see Fig.9). Armed with these new ideas, Hager and his colleagues argued that even Eulerian methods were not fully analyzed and set out to formulate a new theory for approximations in optimal control problems[5]. In other words, the Bernoulli-Euler method had be combined with Karush's generalization of Lagrange's other idea of multipliers using 20^{th} century mathematics of set-valued analysis and convergence in Sobolev spaces – precisely the same tools that were needed to address the discrepancies between the Bellman and Pontryagin theories. Thus, theory and computation were connected in first principles itself! Further, Hager showed that by using the new family of Runge-Kutta methods, one could solve problems by direct methods to the same accuracy as indirect methods. *In other words, there was no longer an accuracy problem with direct and indirect methods. In addition, accurate costates could be obtained by a simple change in coordinates.* The same principles[45] apply to pseudospectral methods[46] as well, and were discovered nearly the same time and refined‖ in subsequent papers[47, 48, 49, 50, 51]. So, we now move from a set of theorems to a single principle as depicted in Fig. 9. For the purposes of completeness, the covector mapping principle is stated as:

Proposition 1 (Covector Mapping Principle) *Let* $[t_0^\infty, t_f^\infty] \mapsto \{\mathbf{x}^\infty, \mathbf{u}^\infty\}$ *be an exact solution to a given optimal control problem* P*. Then, under appropriate conditions, there exists a sequence of inexact problems* $\{P^N\}_{N=0}^{\infty}$ *such that,*

a) $[t_0^\infty, t_f^\infty] \mapsto \{\mathbf{x}^\infty, \mathbf{u}^\infty\}$ *is a solution for Problem* P^N *for all finite* N*, and*

b) Problem P^∞ *= Problem* P*,*

In addition, for all such sequences, there exists

c) time intervals $\left\{[t_0^N, t_f^N]\right\}_{N=0}^{\infty}$*,*

d) system trajectories, $\{t \mapsto (\mathbf{x}^N, \mathbf{u}^N)\}_{N=0}^{\infty}$*,*

d) (covector) functions $\left\{t \mapsto \boldsymbol{\lambda}^N\right\}_{N=0}^{\infty}$*, and*

e) mappings $\{\Gamma^N : \mathbb{R}_{N_x} \to \mathbb{R}_{N_x}\}_{N=0}^{\infty}$*,*

such that for all finite N*,*

1. $[t_0^N, t_f^N] \mapsto \left\{\mathbf{x}^N, \mathbf{u}^N, \boldsymbol{\lambda}^N\right\}$ *satisfies the optimality conditions for Problem* P *inexactly,*

2. $t \mapsto \left\{\mathbf{x}^N, \mathbf{u}^N, \Gamma^N(\boldsymbol{\lambda}^N)\right\}$ *satisfies the inexact optimality conditions for Problem* P^N*,*

and

$$\lim_{N\to\infty} \left\{t \mapsto \Gamma^N(\boldsymbol{\lambda}^N)\right\} = \lim_{N\to\infty} \left\{t \mapsto \boldsymbol{\lambda}^N\right\} = \{t \mapsto \boldsymbol{\lambda}^\infty\}$$

where $[t_0^\infty, t_f^\infty] \mapsto \boldsymbol{\lambda}^\infty$ *together with* $[t_0^\infty, t_f^\infty] \mapsto \{\mathbf{x}^\infty, \mathbf{u}^\infty\}$ *satisfy the exact optimality conditions for Problem* P*.*

All that it needed to make the CMP precise are unambiguous definitions of "appropriate conditions", "inexact optimality conditions," and other terms in the statement of Proposition 1. In adding this precision to the principle, we end up with covector mapping theorems[34, 47]. There are a lot of nuances to this principle but what it implies is that *the easiest way to solve a trajectory optimization problem is to move move counterclockwise in Fig. 9.* That is, to solve a given Problem P, discretize it by any method that satisfies the CMP, and solve the resulting discrete problem for some N sufficiently large. If the covector map, Γ^N, is explicit, it can provide the complete set of multiplier information so that all the necessary conditions of optimality can be verified as if one solved the problem by applying Pontryagin's Principle. *It is important to note what the CMP does not say*:

‖The early results on pseudospectral methods[45] were incomplete.

Remark 1.1 The CMP does not say that solving a sequence of Problems P^N will generate a sequence of solutions for Problem P even if P^N is designed such that $\lim_{N\to\infty} P^N = P$. Although this is a highly desirable result, it is a convergence type theorem that needs to be developed specifically for specific schemes that generate P^N.

Remark 1.2 When P^N is an NLP, note that the CMP does not say that the multipliers of the NLP converge to the multipliers of P. Rather, the CMP states that there exists multipliers for the NLP that converge to the multipliers of P. These convergent sequence of multipliers are given by the mappings Γ^N. The subscripts on $\mathbb{R}$ in statement e) of Proposition 1 essentially emphasize that these are transformations in dual space.

So what happened to the Hamiltonian sensitivity problem? How did it vanish? Clearly if a direct methods equivalent to an indirect method is designed, would it not inherit all the sensitivity issues? It appears it would; but the CMP circumvents it. There are two main reasons for this: In the RK discretizations (including Euler and Hermite-Simpson), the equivalent indirect method is symplectic[6, 34]. This is one reason why the Hager-Runge-Kutta methods are significant. Note that symplecticity does not ensure convergence. The other reason is that the differential equations are not integrated; that is, there is no propagation of the equations.

THE CASE FOR PSEUDOSPECTRAL METHODS

In the late 1990s, a new consensus began to emerge in both the theory and practice for solving optimal control problems. Optimal control theory had found its home in Sobolev spaces[28]; see Ref.[52] for a practical illustration of the utility of Sobolev spaces. Set-valued maps were needed to handle nonsmoothness[16, 28]. Hager's results required both the concepts of set-valued maps and convergence in Sobolev spaces. While Hager's Runge-Kutta methods provide answers and ideas to design numerical methods, they are still only as accurate as the order of the RK method employed. Is this the best we can do? The answer is no. Pseudospectral (PS) methods offer Eulerian-like simplicity while providing very fast convergence rates known as spectral accuracy[50]. For example, PS methods offer exponential convergence rate for analytic functions. Although this fact alone makes the case for PS methods, they are actually the most natural method for solving trajectory optimization problems as they maintain certain geometric connections. To illustrate this point, consider Newton's second law of motion,

$$\ddot{x} = F/m \tag{7}$$

In the correct approach to Newtonian mechanics, we think of $\ddot{x}$ as a separate object from F/m and appreciate that Newton's law brings them together. We do not think of dynamics in terms of integration,

$$x(t) = \int\int \frac{F}{m}\, dt \tag{8}$$

Recursive RK methods view differential equations as something that needs to be integrated. That is, an RK method views dynamics as Eq.(8) and not Eq.(7). Although Eqs.(7) and (8) are mathematically equivalent, there are significant differences between them from the point of view of both physics and computation[6]. A PS method takes the view that a differential equation consists of two separate objects as demanded by physics.** So, the focus of the approximation with RK methods is on the equation and not on the separate sides of the differential equation even though the physics views the left and right-hand sides separately. In contrast, PS methods view a differential equation as a differential equation while an integral is viewed as an integral. That is, in a PS method, it is unnecessary to have an equation to talk about approximations. We talk about approximating derivatives; for example, what is the best way to approximate $\ddot{x}$? This question is independent of whether $\ddot{x}$ is equal to F/m.

**The mathematics of this is described in terms of the tangent bundle and the vector field.

The *coup de grace* to all this discussion is that the natural home of a PS method is a Sobolev space. Given that a Sobolev space is the home of modern optimal control, it is obvious that the most natural way to solve trajectory optimization problems are PS methods that exploit the structure of these Sobolev spaces. This is one reason why PS methods have never failed in solving trajectory optimization problems (assuming they are implemented correctly!). The class of problems that have been solved by PS methods range from minimum-fuel space trajectories[45, 52], spacecraft formation design and control[53, 54], cycler trajectory design[55], sample-return mission design[56], attitude control[57, 58], tether control[59] launch vehicle guidance[60], reentry[61] and many other problems[2, 62, 63]. Most of these problems have been solved by way of the software package, DIDO[64]. DIDO is a minimalist's approach to solving trajectory optimization problem. Only the problem formulation (Problem P) is required in a form that is almost identical to writing it on a piece of paper and pencil. The rest of the process of traversing counterclockwise in Fig.9 is completely transparent to the the user. *Hence, one can solve problems with almost complete disregard to the method of computation and pay attention to only the physics of the problem.*

TYING UP THE LOOSE ENDS: THE $\mathcal{LSA}(P)$ PHILOSOPHY

In going back to the fundamentals, consider once again all methods for solving a trajectory optimization problem. Both the Bellman and Pontragin approaches perform four common steps towards a proposal for constructing solutions for Problem, P:

H Hypothesis: Assume a solution exists for Problem P and its perturbations;

$\mathcal{A}$ Approximate: Perturb the solution and generate various approximations;

$\mathcal{L}$ Take limits: Generate limiting conditions – these are the optimality conditions;

$\mathcal{S}$ Solve: Solve for the limiting conditions.

Unlike the hypothesis step, the last three steps are operations; hence, the process can be summarized as,

$$\mathcal{SLA}(P)$$

Of course, the details of $\mathcal{S}, \mathcal{L}$ and $\mathcal{A}$ in either the Bellman or the Pontryagin approaches are wildly different. Considering that both approaches generate difficult problems at the end, it is reasonable to argue that the commonality of these approaches may be the root cause of the difficulties. In the CMP framework, we commute the last two steps to write,

$$\mathcal{LSA}(P)$$

That is, we solve for the approximate problem and take limits afterwards, exactly the way Bernoulli and Euler approached the problems of their day. *The new theoretical addition to their idea is that we need to commute $\mathcal{L}$ and $\mathcal{S}$ in dual spaces as well.* When this new theoretical addition is combined with modern computing technology, we take limits for convergence analysis but limit the actual computation for a finite N with no prejudice (see Theorem 1). Eulerian methods require the largest N while PS methods require the smallest N to generate the same accuracy. Just as the details of $\mathcal{S}, \mathcal{L}$ and $\mathcal{A}$ operations in the Bellman and the Pontryagin frameworks are different, so is the case in the CMP approach. By postponing the limiting process to the last step, the CMP approach becomes remarkably powerful in much the same way as Pontryagin's Hamiltonian is significantly more powerful than Hamiltion's Hamiltonian[25] given that the only difference between them is a commutative operation. This gives credence to the notion that it was the *path* to problem solving (i.e. the $\mathcal{SLA}(P)$ path) that generated hard subproblems while Problem P itself might have been easy if the right path (i.e. the $\mathcal{LSA}(P)$ path) was chosen.

To draw an analogy between easy and hard methods for solving problems, consider the problem of obtaining equations of motion in mechanics. Newton's method, while attractive and visual (in terms of free-body diagrams), is highly cumbersome for complex systems. On the other hand, Lagrange's method is a "mechanized" easy process whose effectiveness is more pronounced with an increase in the complexity of the dynamical system. Thus, to solve a simple problem like the brachistochrone problem, the $\mathcal{SLA}(P)$ approach (Bellman or Pontryagin) would provide solutions in terms of well-recognizable functions (e.g. cycloid). Now imagine the same problem posed with friction: this minor modification makes the $\mathcal{SLA}(P)$ approach quite undesirable, while the $\mathcal{LSA}(P)$ approach poses no major obstacles to problem solving. Thus, the CMP completes and modernizes a triad of concepts for solving trajectory optimization problems. In using the word, "exact" in a relative sense, it can be said that the $\mathcal{LSA}(P)$ philosophy favors approximate solutions to "exact problems" while the $\mathcal{SLA}(P)$ philosophy favors "exact solutions" to approximate problems.

CONCLUSIONS: THE BREATHTAKING PROSPECTS FOR THE FUTURE

The plot for the story of the CMP is Fig. 9. Contained within it are mathematical details of set-valued analysis and convergence in Sobolev spaces. It is not yet a complete story as it has opened a door to a vast number of new problems from abstract mathematics (e.g. category theory) to engineering. Regardless, the CMP simplifies a broad set of different concepts under a single theme by uniting some apparently disparate topics in analysis.

That a large number of trajectory optimization problems can be easily solved, and many in real time as well, is a clear modern-day reality. That is, even if advances in computational hardware and algorithms came to complete standstill, it is still possible to advance faster solutions to complex trajectory problems by simply integrating existing tools. Just as Lagrangian dynamics did not kill the field of dynamics, but rather enhanced it, the CMP does not the kill the field of trajectory optimization just because many difficult problem are rendered easy. In fact, as a result of the explosive growth in new concepts, a vast number of new research areas have emerged. Delineating these new topics would require another paper.

ACKNOWLEDGMENTS

I would like to express my deep gratitude to my colleagues, Profs. Fariba Fahroo and Wei Kang, and to Dr. Qi Gong for developing rigorous proofs to the many theorems I have utilized in this paper in the form of assertions.

References

[1] http://en.wikipedia.org/wiki/Egg_of_Columbus (Wikipedia, The Free Encyclopedia).

[2] Ross, I. M. and D'Souza, C. D., "Hybrid Optimal Control Framework for Mission Planning," *Journal of Guidance, Control and Dynamics*, Vol. 28, No. 4, July-August 2005, pp. 686-697.

[3] Boggs, P. T., Kearsley, A. J., and Tolle, J. W., "A Global Convergence Analysis of an Algorithm for Large-Scale Nonlinear Optimization Problems," *SIAM Journal of Optimization*, Vol. 9, No. 4, 1999, pp. 833-862.

[4] Rockafellar, R. T. "Lagrange Multipliers and Optimality," *SIAM Review*, 35 1993, pp. 183-238.

[5] Hager, W. W., "Numerical Analysis in Optimal Control," *International Series of Numererical Mathematics*, Hoffmann, K.-H. Lasiecka, I., Leugering, G., Sprekels, J., and Troeltzsch, F., Eds., Birkhäuser, Basel, Switzerland, 2001, Vol. 139, pp. 83–93.

[6] Ross, I. M., "A Roadmap for Optimal Control: The Right Way to Commute," *Annals of the New York Academy of Sciences*, to appear.

[7] Mordukhovich, B. S. and Shvartsman, I., "The Approximate Maximum Principle in Constrained Optimal Control," *SIAM Journal of Control and Optimization*, Vol. 43, No. 3, 2004, pp. 1037-1062.

[8] Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics,* Vol. 21, No. 2, 1998, pp. 193-207.

[9] Mordukhovich, B. S., *Variational Analysis and Generalized Differentiation, I: Basic Theory*, vol. 330 of Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences] Series, Springer, Berlin, 2005.

[10] Mordukhovich, B. S., *Variational Analysis and Generalized Differentiation, II: Applications*, vol. 331 of Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences] Series, Springer, Berlin, 2005.

[11] Bryson, A. E., and Ho, Y.-C., *Applied Optimal Control,* Hemisphere, New York, 1975 (Revised Printing; original publication, 1969).

[12] Ferris, M. C. and Munson, T. S., "Interior-Point Methods For Massive Support Vector Machines," *SIAM Journal of Optimization*, Vol. 13, No. 3, 2003, pp. 783-804

[13] Nahin, P. J., *When Least is Best*, Princeton University Press, Princeton, NJ, 2004.

[14] http://www-history.mcs.st-and.ac.uk/history/HistTopics/Brachistochrone.html

[15] Sussmann, H. J. and Willems, J. C., "The Brachistochrone Problem and Modern Control Theory," *Contemporary Trends in Nonlinear Geometric Control Theory and its Applications*, A. Anzaldo-Meneses, B.Bonnard, J.-P. Gauthier, and F. Monroy-Perez (Eds); World Scientific Publishers, Singapore, 2000.

[16] Clarke, F. H., Ledyaev, Y. S., Stern, R. J., and Wolenski, P. R., *Nonsmooth Analysis and Control Theory,* Springer-Verlag, New York, NY, 1998.

[17] Adamchik V., and Wagon, S., "Pi: A 2000-Year Search Changes Direction," *Education and Research*, Vol. 5, No.1, 1996, pp. 11-19.

[18] Kolmogorov, A. N. and Fomin, S. V., *Elements of the Theory of Functions and Functional Analysis*, Vol. 2, Dover Publications, Mineola, N.Y., 1999.

[19] Andraka, R., "A Survey of CORDIC Algorithms for FPGAs," *Proceedings of the 1998 ACM/SIGDA Sixth International Symposium on Field Programmable Gate Arrays*, ACM, Inc., Feb. 22-24, 1998, Monterey, CA, pp. 191-200.

[20] Goldstine, H. H., *A History of the Calculus of Variations from the 17th to the 19th Century*, Springer-Verlag, New York, N.Y., 1981, pg. 110.

[21] Duren, W. L., Jr., "Graduate Student at Chicago in the Twenties," *The American Mathematical Monthly*, Vol. 83, No. 4. (Apr., 1976), pp. 243-248.

[22] Kuhn, H. W., "Being in the Right Place at the Right Time," *Operations Research*, Vol. 50, No. 1, JanuaryFebruary 2002, pp. 132134.

[23] Dreyfus, S., "Richard Bellman on The Birth of Dynamic Programming," *Operations Research*, Vol.50, No.1, Jan-Feb 2002, pp.48-51

[24] Gamkrelidze, R. V., "Discovery of the Maximum Principle," *Journal of Dynamical and Control Systems*, Vo. 5, No. 4, 1999, pp. 437-451.

[25] Sussmann, H. J. and Williems, J. C., "300 Years of Optimal Control: From the Brachystochrone to the Maximum Principle," *IEEE Control Systems Magazine*, June 1997, pp. 32-44.

[26] Vinter, R., "Review of, 'Dynamic Optimization,' by A. E. Bryson," *Automatica*, Vol. 38, pp. 1831-1833, 2002.

[27] Bardi, M. and Capuzzo-Dolcetta, I., *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*, Birkhäuser, Boston, MA, 1997.

[28] Vinter, R. B., *Optimal Control*, Birkhäuser, Boston, MA, 2000.

[29] Lawden, D. F., *Optimal Trajectories for Space Navigation*, Butterworths, London, 1963.

[30] Marec, J.-P., *Optimal Space Trajectories*, Elsevier, New-York, 1979.

[31] Halkin, H., "A Maximum Principle of the Pontryagin Type for Systems Described by Nonlinear Difference Equations," *SIAM Journal of Control*, Vol. 4, No. 1, 1966, pp. 90-111.

[32] Cullum, J., "Discrete Approximations To Continuous Optimal Control Problems," *SIAM Journal of Control*, Vo1.7, No. 1, February 1969, pp. 32-49.

[33] Chang, S. S. L., "On Convexity and the Maximum Principle for Discrete Systems," *IEEE Transactions on Automatic Control*, Vol. AC-11, Feb. 1966, pp.121-123.

[34] Hager, W. W., "Runge-Kutta Methods in Optimal Control and the Transformed Adjoint System," *Numerische Mathematik,* Vol. 87, 2000, pp. 247-282.

[35] Hargraves, C. R. and Paris, S. W., "Direct Trajectory Optimization Using Nonlinear Prgroamming and Collocation," *Journal of Guidance, Control and Dynamics*, Vol.10, 1987, pp.338-342.

[36] Dickmanns, E. D. and Well, K. H., "Approximate Solution of Optimal Control Problems Using Third-Order Hermite Polynomial Functions," *Proceedings of the 6th Technical Conference on Optimization Techniques*, Springer-Verlag, New York, IFIP-TC7, 1975.

[37] Polak, E., "An Historical Survey of Computational Methods in Optimal Control," *SIAM Review* Vol. 15, No. 2, April 1973, pp.553-584.

[38] Hager, W. W., "Rate of Convergence for Discrete Approximations to Unconcstrained Control Problems," *SIAM Journal of Numerical Analysis*, Vol.13, 1976, pp.449-471.

[39] Paris, S. W. and Hargraves, C. R., *OTIS 3.0 Manual*, Boeing Space and Defense Group, Seattle, WA, 1996.

[40] Betts, J. T., *Practical Methods for Optimal Control Using Nonlinear Programming,* SIAM: Advances in Control and Design Series, Philadelphia, PA, 2001.

[41] Herman, A. L. and Conway, B. A., "Direct Optimization Using Collocation Based on High-ORder Gauss-Loabatto Quadrature Rules," *Journal of Guidance, Control, and Dynamics,* Vol. 19, No. 3, 1996, pp. 592-599.

[42] Enright, P. G. and Conway B. A., "Discrete Approximations to Optimal Trajectories Using Direct Transcription and Nonlinear Programming," *Journal of Guidance, Control, and Dynamics,* Vol. 15, No. 4, 1992, pp. 994-1002.

[43] von Stryk, O., "Numerical Solution of Optimal Control Problems by Direct Collocation," *Optimal Control: Calculus of Variations, Optimal Control Theory and Numerical Methods*, Edited by Bulirsch, R., et al., Birkhäuser, 1993.

[44] Ross, I. M. and Fahroo, F., "A Perspective on Methods for Trajectory Optimization," *Proceedings of the AIAA/AAS Astrodynamics Conference*, Monterey, CA, August 2002. AIAA Paper No. 2002-4727.

[45] Fahroo, F. and Ross, I. M., "Costate Estimation by a Legendre Pseudospectral Method," *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 10-12 August 1998, Boston, MA; also in *Journal of Guidance, Control and Dynamics*, Vol.24, No.2, March-April 2001, pp.270-277.

[46] Elnagar, J., Kazemi, M. A. and Razzaghi, M., "The Pseudospectral Legendre Method for Discretizing Optimal Control Problems," *IEEE Transactions on Automatic Control,* Vol. 40, No. 10, 1995, pp. 1793-1796.

[47] Ross, I. M. and Fahroo, F., "A Pseudospectral Transformation of the Covectors of Optimal Control Systems," *Proceedings of the First IFAC Symposium on System Structure and Control*, Prague, Czech Republic, 29-31 August 2001.

[48] Ross, I. M. and Fahroo, F., "Legendre Pseudospectral Approximations of Optimal Control Problems," *Lecture Notes in Control and Information Sciences*, Vol.295, Springer-Verlag, New York, 2003.

[49] Ross, I. M. and Fahroo, F., "Discrete Verification of Necessary Conditions for Switched Nonlinear Optimal Control Systems," *Proceedings of the American Control Conference*, June 2004, Boston, MA.

[50] Gong, Q., Ross, I. M., Kang, K. and Fahroo, F., "Convergence of Pseudospectral Methods for Constrained Nonlinear Optimal Control Problems," *Proceedings of the IASTED International Conference on Intelligent Systems and Control*, Honolulu, HI, pp. 209–214, 2004.

[51] Ross, I. M. and Fahroo, F. and Gong, Q., "A Spectral Algorithm for Pseudospectral Methods in Optimal Control," *Proceedings of the 10th International Conference on Cybernetics and Information Technologies, Systems and Applications (CITSA)*, July 21-25, 2004, Orlando, FL, pp. 104-109.

[52] Ross, I. M., "How to Find Minimum-Fuel Controllers," *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Providence, RI, August 2004. AIAA Paper No. 2004-5346.

[53] Ross, I. M., King, J. T., Fahroo, F., "Designing Optimal Spacecraft Formations," *Proceedings of the AIAA/AAS Astrodynamics Conference*, AIAA-2002-4635, Monterey, CA, 5-8 August 2002.

[54] Infeld, S. I., Josselyn, S. B., Murray W. and Ross, I. M., "Design and Control of Libration Point Spacecraft Formations," *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Providence, RI, August 2004. AIAA Paper No. 2004-4786.

[55] Stevens, R. and Ross, I. M., "Preliminary Design of Earth-Mars Cyclers Using Solar Sails," to appear in the *Journal of Spacecraft and Rockets*; also, *AAS/AIAA Spaceflight Mechanics Conference*, Paper No. AAS 03-244, Ponce, Puerto Rico, 9-13 February 2003.

[56] Stevens, R., Ross, I. M., and Matousek, S. E., "Earth-Mars Return Trajectories Using Solar Sails," *55th International Astronautical Congress*, Vancouver, Canada. Paper IAC-04-A.2.08.

[57] Sekhavat, P., Fleming A. and Ross, I. M., "Time-Optimal Nonlinear Feedback Control for the NPSAT1 Spacecraft," *Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, AIM 2005, 2428 July 2005 Monterey, CA.

[58] Yan, H., Lee, D. J., Ross, I. M. and Alfriend, K. T., "Real-Time Outer and Inner Loop Optimal Control Using DIDO," *AAS/AIAA Astrodynamics Specialist Conference*, Tahoe, NV, August 8-11, 2005, Paper AAS 05 - 353.

[59] Williams, P.. Blanksby C. and Trivailo, P., "Receding Horizon Control of Tether System Using Quasilinearization and Chebyshev Pseudospectral Approximations," *AAS/AIAA Astrodynamics Specialist Conference*, Big Sky, MT, August 3-7, 2003, Paper AAS 03-535.

[60] Rea, J., "Launch Vehicle Trajectory Optimization Using a Legendre Pseudospectral Method," *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Austin, TX, August 2003. Paper No. AIAA 2003-5640.

[61] Josselyn S. and Ross, I. M., "A Rapid Verification Method for the Trajectory Optimization of Reentry Vehicles," *Journal of Guidance, Control and Dynamics*, Vol. 26, No. 3, 2003.

[62] Ross, I. M. and Fahroo, F., "Issues in the Real-Time Computation of Optimal Control," *Mathematical and Computer Modelling*, Vol. 40, Pergamon Publication (to appear).

[63] Mendy, P. B., "Multiple Satellite Trajectory Optimization," Astronautical Engineer Thesis, Department of Mechanical and Astronautical Engineering, Naval Postgraduate School, Monterey, CA, December 2004.

[64] Ross, I. M., *User's Manual for DIDO: A MATLAB Application Package for Solving Optimal Control Problems*, Technical Report 04-01.0, Tomlab Optimization Inc, February 2004.